Índice

Capítulo 1 Introducción a la Visión Artificial

En este tema se introducen una serie de conceptos físicos y fisiológicos imprescindibles para entender el por qué de muchas decisiones de diseño que se toman al construir los sistemas de visión computacional.

1.1. Modelo físico de la luz

Desde el punto de vista del procesado de imagen, basta considerar la luz como una onda. Según este modelo las características de un rayo de luz vienen completamente determinadas por dos propiedades: su amplitud y su longitud de onda. Sin embargo este modelo para explicar la luz no es ni el primero, ni tampoco, el que más se aproxima a las observaciones.

1.1.1 La luz en la historia

La primera teoría sobre la naturaleza de la luz fue probablemente debida a Euclides (330 antes de Cristo) que suponía que la luz era una especie de rayo lanzado por el ojo hacia la cosa vista. Esta teoría tiene diversos errores, quizás el más patente consiste en que no explica por qué no se puede ver en ausencia de luz.

Ya en el año 1000 de nuestra era, el árabe Alhacen afirmó que la luz se dirige desde una fuente que la emite hasta nuestros ojos, después de ser reflejada

por los objetos vistos. Pero fue en el siglo XVII cuando se realizaron los mayores progresos de la mano de Newton, que hizo notables avances en la teoría del color y la dispersión. Newton era defensor de una *teoría corpuscular*, según la cual la luz estaba formada por un flujo de partículas proyectadas por el cuerpo luminoso. Al mismo tiempo, otros científicos, como Hooke y Huygens, defendían una teoría ondulatoria que explicaba mejor ciertos hechos como que dos haces luminosos se cruzasen sin perturbarse. El principal problema de aquella teoría ondulatoria estribaba en que no existía ninguna evidencia empírica del medio en el que se propagaba esa onda. Este medio, que se denominó *éter*, debería llenar el espacio, y debería tener una dureza altísima para permitir la alta velocidad de propagación que caracteriza a la luz. Por ello, y también quizás debido al peso de la autoridad de Newton, la teoría corpuscular se impuso durante doscientos años.

En el siglo XIX, los trabajos de Young, Fresnel y Foucault salvaron la mayoría de las objeciones propuestas por Newton a la *teoría ondulatoria* cosechando numerosos éxitos en el campo de la óptica. El impulso definitivo a favor de la naturaleza ondulatoria de la luz lo dio Maxwell en 1873 con su teoría electromagnética. Ésta explicaba la luz como una radiación de naturaleza ondulatoria que se puede propagar en el vacío, haciendo innecesaria la idea del éter.

Sin embargo, hacia 1900 se constató un nuevo fenómeno, denominado *efecto fotoeléctrico*, que proporcionó evidencias experimentales de que la luz tenía carácter corpuscular en su interacción con la materia. Esto llevó de nuevo al replanteamiento de la naturaleza de la luz. Albert Einstein, Louis de Broglie y otros construyeron una nueva teoría que llamaron *teoría onda-corpúsculo*. Esta teoría considera la luz formada por unas partículas, los fotones, cada una de las cuales tiene asociada una ecuación de ondas. Así, cuando la luz interactúa con la materia, como en el efecto fotoeléctrico, se invoca a un modelo corpuscular para explicar tal interacción, mientras que para explicar fenómenos relativos a su propagación, como en la difracción de los rayos X, se recurre a un modelo ondulatorio. Este modelo dual, onda-corpúsculo, permite explicar la totalidad de los fenómenos observados hasta la fecha.

1.1.2 Definiciones

Aunque las ondas luminosas constituyen una parte muy pequeña del conjunto de ondas electromagnéticas, son especialmente interesantes porque tienen la particularidad de que son captadas por los ojos y procesadas en el cerebro. El ojo

humano es capaz de distinguir radiaciones de longitudes de onda comprendidas entre 400 y 700 nanómetros (1 nanómetro = 10^{-9} metros).

Nuestro sistema sensorial visual interpreta las diferentes amplitudes y longitudes de onda de la luz, produciendo las sensaciones que conocemos como brillo y color respectivamente. Así por ejemplo, una onda electromagnética que viaja por el vacío con una longitud de onda predominante de 680 nanómetros se interpreta en el cerebro como la sensación del color rojo.

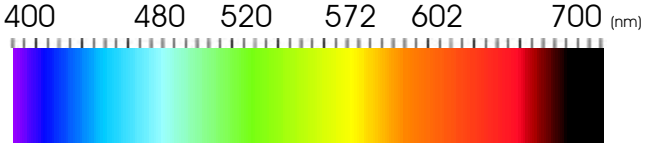


Figura 1.- La parte de la radiación electromagnética que constituyen las ondas luminosas abarca desde el fin del ultravioleta 400 nm hasta el comienzo del infrarrojo 700 nm.

Distribución espectral de energía

La *distribución espectral de energía* es una curva que representa la cantidad de energía (en vatios) asociada a cada longitud de onda en una radiación electromagnética.

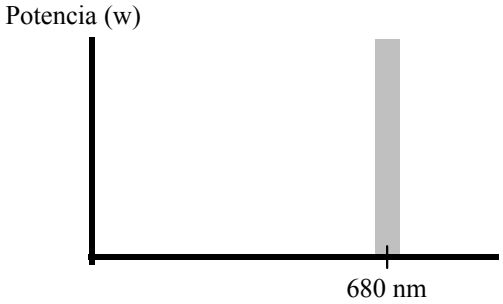


Figura 2.- Diagrama espectral ideal de una luz roja.

Si se representa el diagrama espectral de una radiación electromagnética que posee una longitud de onda igual a 680 nm, se obtiene un gráfico con un pico en la longitud correspondiente a 680 nm y 0 en el resto (ver Figura 2). Una luz de estas características, compuesta por una radiación con una longitud de onda determinada, se denomina luz *monocromática*.

En general, las radiaciones no suelen ser tan puras y resultan de la mezcla de diferentes haces con diferentes longitudes de onda. Así, los diagramas espectrales son más parecidos a los de la Figura 3.

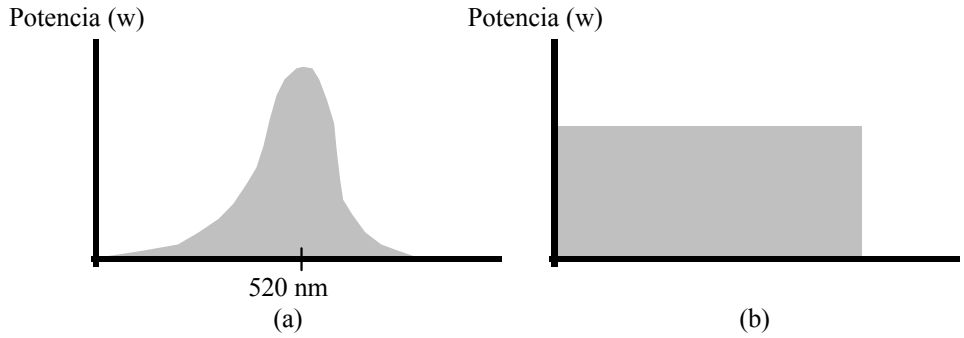


Figura 3.- Diagramas espectral de una luz verde (a) y de una luz blanca (b).

Flujo radiante

El *flujo radiante* es la cantidad de energía emitida por una fuente de ondas electromagnéticas por unidad de tiempo y se mide en vatios (ver Figura 4).

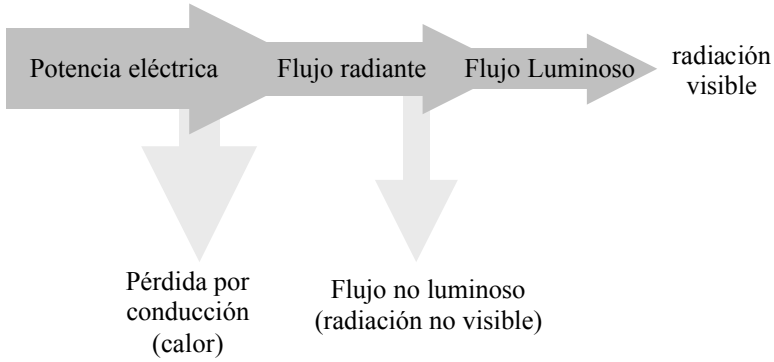


Figura 4.- De la energía usada para producir luz, el humano sólo percibe una pequeña parte que se denomina flujo luminoso.

Flujo luminoso

El *flujo luminoso* es la parte del flujo radiante detectada por el ojo (ver Figura 4). La unidad de flujo luminoso es el *lumen* (L). Un lumen es el flujo luminoso procedente de una abertura de 1/60 cm^2 en un cilindro de material refractario que contiene un material patrón que radia a través de un cono de radiación de un estereorradián. El flujo luminoso se puede medir con un fotómetro y se representa con el símbolo Φ.

Mediante experimentación se ha definido la curva de la Figura 5, que permite obtener, en lúmenes, el flujo luminoso correspondiente a una luz monocromática de cualquier longitud de onda que tenga un flujo radiante igual a un vatio. De este diagrama se deduce, por ejemplo, que una luz monocromática de 1 vatio de potencia, de 600 nm produce una sensación de luminosidad en el ojo humano igual a 420 lúmenes. Además, en él se aprecia que el mayor rendimiento de flujo luminoso se obtiene para las longitudes de onda correspondientes a los tonos verdes.

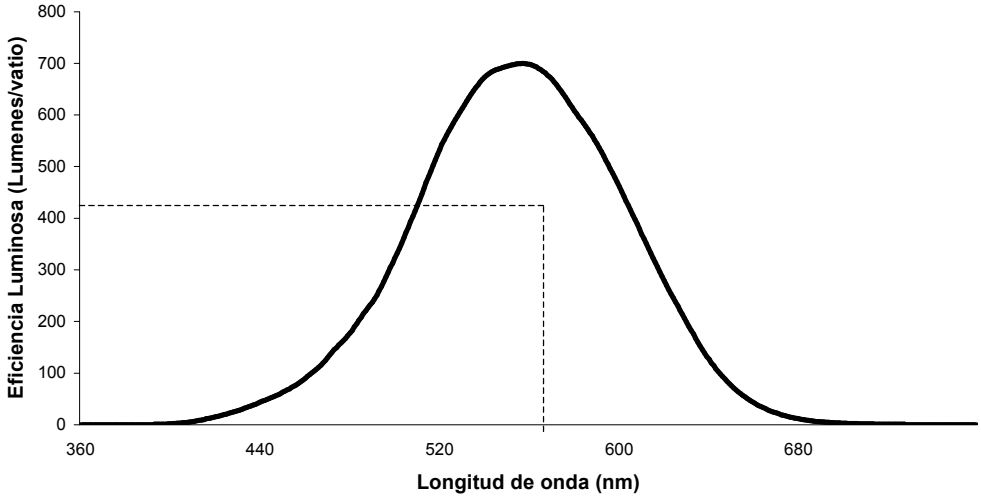


Figura 5.- Esta curva expresa el rendimiento luminoso de un flujo monocromático en función de la longitud de onda.

Esta curva, que llamaremos $V(l)$, permite definir la relación (1.1). Ésta permite calcular el flujo luminoso de una radiación cuando se conoce su distribución espectral $P(l)$.

$$\Phi = \int_0^\infty P(l) \cdot V(l) \cdot dl \quad \text{(L)} \tag{1.1}$$

Ejemplo 1.-

El flujo luminoso, en lúmenes, de una energía radiante de 27 vatios de una fuente luminosa monocromática con una longitud de onda de 600 nm se puede calcular usando el diagrama de la Figura 5. Sobre éste se ve que una luz monocromática de 600 nm produce un flujo luminoso de 420 L. En este caso se tiene que tanto $V(l)$ como $P(l)$ son funciones constantes que toman valores:

$V(l) = 420$ (L/w) $\qquad\qquad$ $P(l) = 27$ (w)

Sustituyendo en (1.1) se obtiene:

$$\Phi = \int_0^\infty P(l) \cdot V(l) \cdot dl = 27 \times 420 = 11340 \quad \text{(L)}$$

Intensidad luminosa

La *intensidad luminosa* (1.2) es el flujo luminoso emitido por unidad de ángulo sólido (Figura 6). Se representa como I, y su unidad es la *bujía* (b), que se corresponde a un lumen/estereorradián.

$$I = \frac{d\Phi}{dw} \quad \text{(b)} \tag{1.2}$$

Luminancia o brillo

La *luminancia* o *brillo* de un manantial es la intensidad luminosa por unidad de superficie.

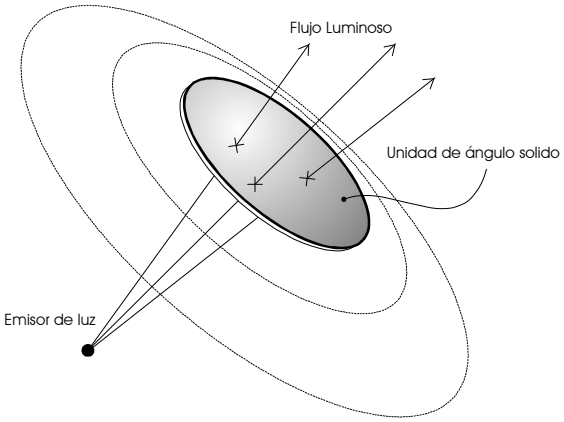


Figura 6.- Representación del flujo luminoso que atraviesa un estereorradián.

1.2. Modelo Fisiológico

Hasta ahora se ha hablado de la luz desde un punto de vista físico. Sin embargo, la correspondencia entre los fenómenos físicos y lo que se percibe no es directa. Hay experimentos que demuestran que a veces nuestro sistema de percepción confunde elementos que son iguales, y a veces encuentra diferencias entre elementos que son idénticos. Esto se debe a que nuestro sistema visual impone ciertas limitaciones que se analizarán en este apartado.

El ojo es un órgano que captura la luz y la transforma en un impulso neuronal que transmite al cerebro para su procesamiento. La luz, tras atravesar una lente llamada cristalino, incide en la retina, que está situada en la parte anterior del ojo. Allí, la luz actúa sobre unas células fotosensibles (los conos y los bastones) que forman parte de la retina. Estas células en presencia de luz generan impulsos neuronales que se envían al cerebro mediante el nervio óptico. El cerebro procesa la información que recibe y genera sensaciones: es lo que se conoce como el proceso de *percepción visual*. A continuación se describe la percepción acromática (sólo teniendo en cuenta el brillo, es decir la cantidad de energía), y después la percepción cromática (que tiene en cuenta el color, esto es, la forma de distribuirse la energía en distintas longitudes de onda).

1.2.1 Percepción acromática

La percepción del brillo de una imagen la realizan en el ojo los bastones (ver Figura 7). Son unas células especializadas que tenemos en la retina, en un número superior a 100 millones, que son capaces de detectar y medir el brillo de los haces luminosos que les llegan. La sensación de brillo está relacionada con dos fenómenos:

- La *sensibilidad a la intensidad*.
- La *inhibición lateral*.

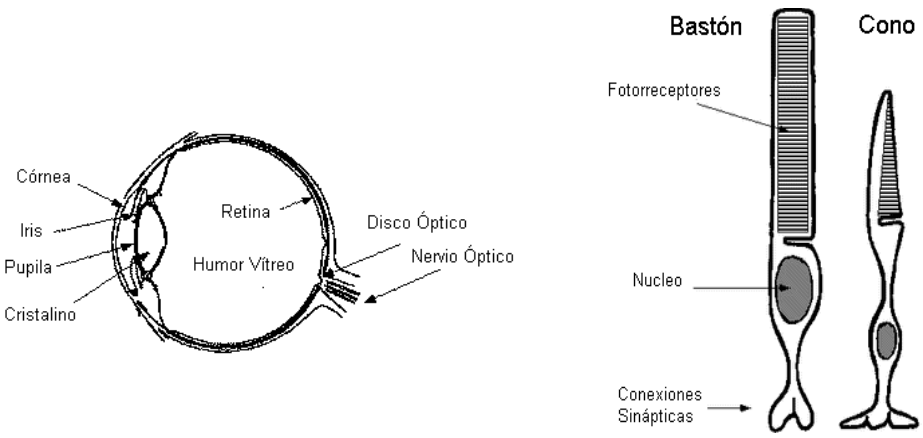


Figura 7.- A la izquierda el ojo con detalle de sus partes más importantes. A la derecha un esquema de un bastón y de un cono.

Sensibilidad a la intensidad luminosa y el contraste

La sensibilidad a la intensidad es lo que dota de la capacidad de distinguir un nivel de intensidad de otro. La diferencia de intensidad se denomina *contraste*. Se dice que una imagen tiene gran contraste si las diferencias de intensidad que contiene son pronunciadas.

La sensibilidad a la intensidad en el ser humano es alta siempre que los elementos que se comparan son pocos. Cuando el número de intensidades involucradas simultáneamente es superior a unos 24 tonos se pierde la mayor parte

de esta sensibilidad. Esto implica que, en la mayoría de casos prácticos, sea suficiente el uso de 32 ó 64 niveles de intensidad para representar una imagen.

Los seres humanos son capaces de distinguir un rango muy amplio de intensidades. Sin embargo la relación entre la intensidad real de la luz reflejada por un pigmento y la intensidad percibida por un humano no es lineal. La curva *A* de la Figura 8 representa el brillo apreciado en relación con el brillo físico reflejado por un pigmento. Se aprecia que el humano es capaz de distinguir pigmentos de intensidades poco diferentes (como el a_1 y el a_2) cuando los cuerpos que tienen esos pigmentos están próximos espacialmente. Sin embargo, los contrastes acentuados hacen que esta sensibilidad decrezca. Por ello, cuando hay involucradas pigmentos con intensidades muy dispares simultáneamente, como el *b* y el *c*, la distinción entre intensidades próximas decrece. De manera que la percepción de a_2 y a_1 se sitúa en curvas similares a la B y a la C respectivamente, que como se aprecia las hacen percibir como lejanas (a_1' y a_2').

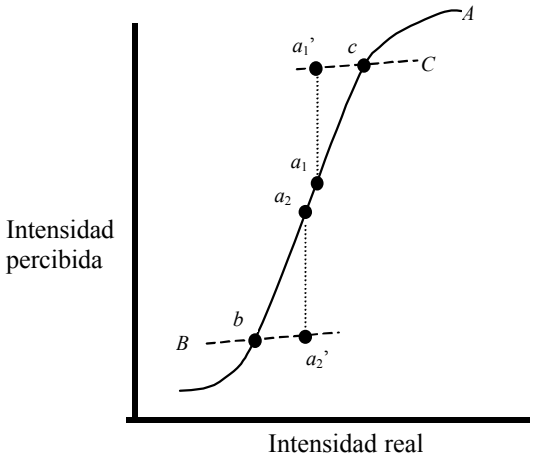


Figura 8.- La línea *A* representa la relación entre el brillo distinguido por el ojo humano y el nivel de brillo real.

En la Figura 9 se puede comprobar este efecto. En ella se percibe que los rectángulos interiores tienen intensidades distintas, cuando en realidad tienen la misma. Se concluye que, aunque el sistema visual es capaz de percibir y distinguir un amplio rango de niveles de brillo, disminuye su precisión cuando hay muchos niveles involucrados a la vez, necesitando periodos de adaptación a cada situación.

Figura 9.- El color gris del cuadrado interior de la figura de la derecha parece más oscuro que el cuadrado interior de la figura de la izquierda, a pesar de que ambos están tintados con el mismo gris.

Inhibición lateral

El otro fenómeno que se indicaba, la inhibición lateral, se origina en el hecho de que las células de la retina, al detectar un nivel de intensidad, inhiben las células vecinas, produciendo perturbaciones en las fronteras de cambio de intensidad. Este fenómeno, que puede apreciarse en la Figura 10, también influye en que el brillo percibido no esté en proporción directa con el brillo físico.

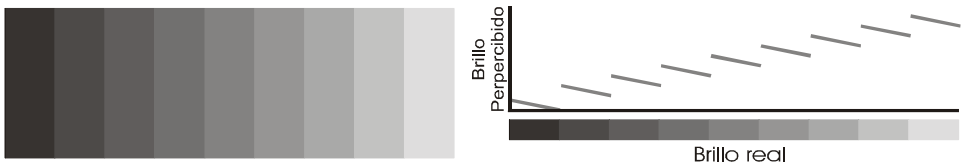


Figura 10.- La tonalidad de cada una de las franjas verticales de la figura de la izquierda es uniforme. Sin embargo, al observarlas, parece que son más oscuras por la derecha y más claras por la izquierda. El brillo percibido para cada banda se refleja en el diagrama de la derecha.

1.2.2 Percepción cromática

La percepción del color de una imagen la realizan los conos (ver Figura 7). Son unas células especializadas, dispuestas en la retina en un número cercano a los 6 millones, que son capaces de variar su comportamiento ante cambios en la longitud de onda de una radiación electromagnética. Basándose en la información aportada por los conos el cerebro construye la sensación de color.

Los conos tienen una sensibilidad menor que los bastones. Se dice popularmente que *"de noche todos los gatos son pardos"*, reflejando el hecho de que con poca luz sólo los bastones captan suficiente energía para activarse.

La sensación de color que percibimos está relacionada con la energía que tiene a diferentes longitudes de onda una radiación electromagnética. Para explicar tal relación se definen el *matiz* y la *saturación*.

Matiz

El matiz o *tono* se deriva de la relación que se produce entre las activaciones de los distintos tipos de conos cuando sobre ellos incide la luz. El matiz depende de la longitud de onda dominante, es decir, aquélla para la que se encuentra más energía en el diagrama espectral (ver Figura 11). El ser humano es capaz de distinguir entre 125 y 150 matices distintos cuando están próximos, perdiendo esa capacidad si están distanciados espacialmente.

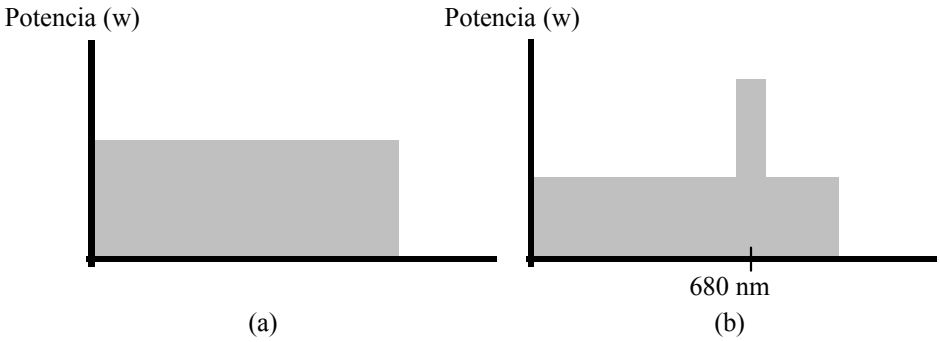


Figura 11.- La figura de la izquierda no tiene una longitud de onda dominante, su matiz es blanco. La figura de la derecha corresponde a un objeto rojo, siendo la longitud de onda dominante la correspondiente a 680 nm.

Saturación

Mide la proporción entre la longitud de onda dominante y el resto de longitudes de onda. En la Figura 12 se presenta un ejemplo de dos diagramas espectrales con el mismo matiz, pero con diferente saturación.

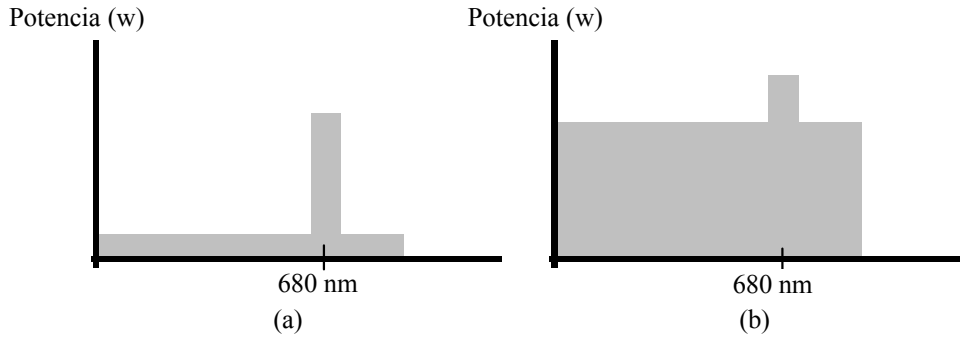


Figura 12.- Dos espectros con el mismo matiz. El de la izquierda corresponde a un rojo muy saturado. El de la derecha a una luz roja poco saturada.

Definidos los conceptos de matiz, saturación y brillo se dice que se ve un color determinado cuando se percibe una cierta combinación de estos tres elementos.

Tipos de conos

Estudios fisiológicos han revelado que existen tres tipos de conos, denominados tipos S, L, y M. Los conos de tipo S (short) son más sensibles a las radiaciones con longitud de onda corta (azules), los M (medium) a las radiaciones de longitud media (verdes), y los L (large) a las de longitud larga (rojos). El hecho de la existencia de sólo tres tipos de receptores, para percibir todos los colores, es la base de la *teoría triestímulo*.

1.2.3 Diagrama cromático y teoría triestímulo

La mezcla aditiva de la luz emitida por tres linternas, una roja (de 650 nm), otra verde (de 530 nm) y otra azul (de 425 nm) permite obtener una amplia gama de colores. Sobre este hecho se sustentan multitud de dispositivos que generan imágenes en color, como los tubos de los televisores y monitores, las pantallas de cristal líquido, las pantallas de plasma, etc.

Sin embargo, se ha demostrado que no es posible obtener todos los colores que un humano puede distinguir mediante mezcla aditiva de tres linternas[1]. Estos resultados se encuentran dentro de la denominada *teoría triestímulo*.

En 1931 la Comisión Internacional de Iluminación (C.I.I.) convino expresar qué cantidad, en Lúmenes, tiene que emitir cada una de tres linternas patrón para expresar todos los matices del espectro de manera aditiva (ver Figura 13). Los colores de estas tres linternas, que se denominaron a, b y c, se encuentran fuera del dominio de los colores reales, pero esto carece de importancia, puesto que las cantidades luz de estas linternas necesarias para igualar cualquier matiz del espectro, se calculan por métodos matemáticos.

Para obtener un color determinado se toma A como la cantidad del componente a, B como la cantidad del componente b, y C como la cantidad del componente c. Ahora, con objeto de normalizar A, B y C, se plantean las siguientes relaciones:

$$x = \frac{A}{A+B+C}, \quad y = \frac{B}{A+B+C}, \quad z = \frac{C}{A+B+C}$$

Evidentemente $x+y+z=1$, con lo que el valor de z depende de los valores de x e y. Por ello sólo son necesarias las magnitudes x e y para definir cualquier color. Representando en un plano XY el color asociado a cada punto (x,y,z) se obtiene el gráfico de la Figura 14 (a).

[1] Aunque no es posible obtener todos los colores mediante mezcla aditiva de tres linternas, sí es posible expresar matemáticamente todos los colores como combinación lineal de tres linternas. La mezcla aditiva puede realizarse en la realidad sin más que mezclar la luz de las linternas. La combinación lineal debe permitir la resta de luces, cosa que no es posible físicamente.

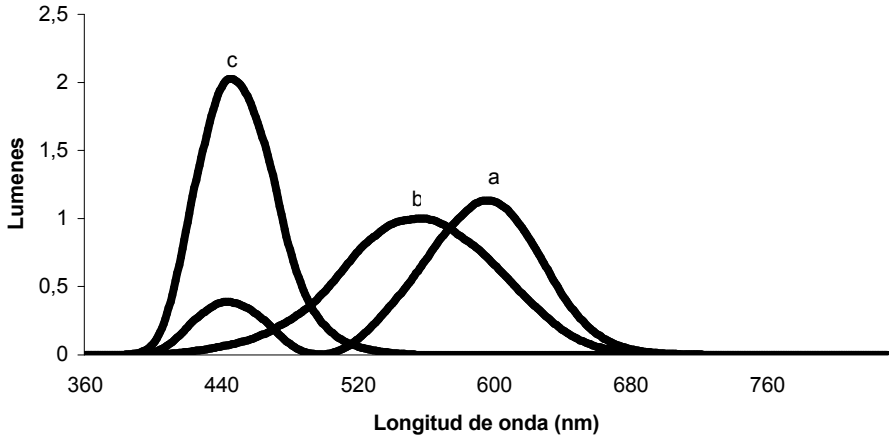
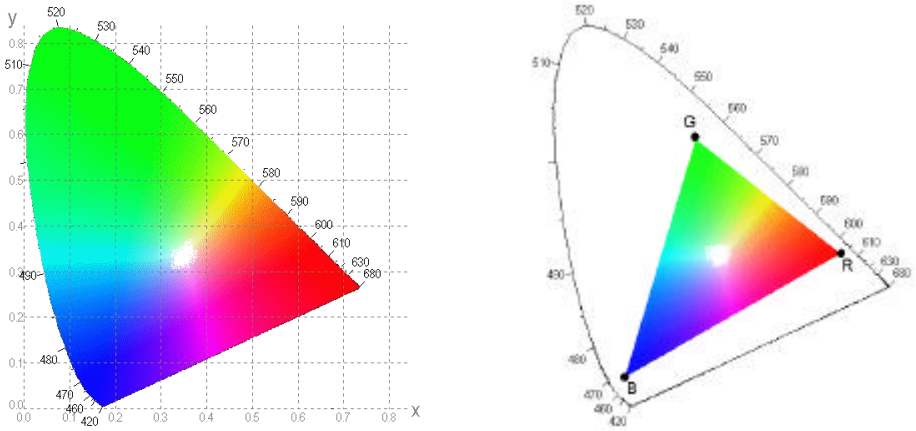


Figura 13.- Curvas fijadas por la C.I.I., mediante experimentos estadísticos con personas, que fijan el número de lúmenes de cada una de tres linternas monocromáticas necesarias para igualar un vatio de flujo radiante para cada matiz del espectro.



(a) (b)

Figura 14.- Diagrama Cromático del C.I.I.

Los colores comprendidos dentro del triángulo definido por los puntos R, G y B[2] de la Figura 14 (b) son aquéllos que se pueden obtener por mezcla aditiva de tres linternas con los colores correspondientes a los vértices del triángulo. Por eso los televisores y los monitores de ordenador, eligen cuidadosamente la base de tres colores (rojo, verde y azul) que usarán para construir las imágenes que presentan, de manera que el área del triangulo dentro del diagrama C.I.I. sea máxima, y así poder representar el máximo posible de colores en sus pantallas.

El diagrama C.I.I. da lugar a otro tipo de representación denominado *HSV* (Matiz Saturación y luminosidad[3]). Esta representación puede considerarse como un superconjunto de una representación RGB. Una descripción más amplia de los modelos de representación del color puede encontrarse en la bibliografía que aparece al final del capítulo.

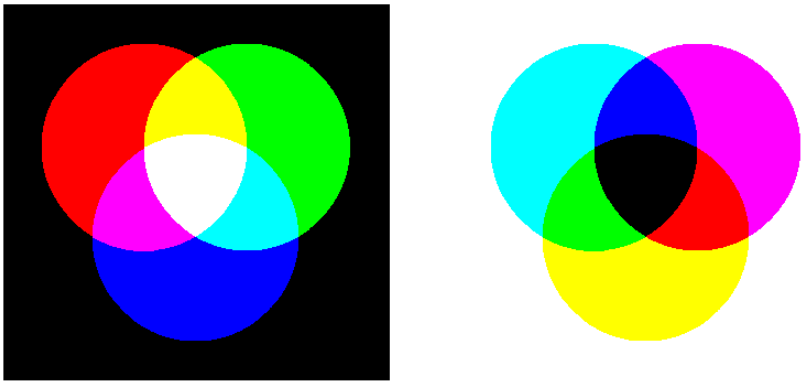


Figura 15.- A la izquierda mezcla aditiva de la luz tres linternas sobre una superficie blanca no iluminada. A la derecha mezcla substractiva de tres tintes sobre un lienzo blanco.

Tintas

Usando tintas no se puede realizar la mezcla aditiva de haces de luz, ya que los colores percibidos al mezclar tintas son el resultado de una mezcla substractiva. Así, al tintar un papel blanco se le van restando componentes que ya no podrá

[2] *RGB* por red, green, blue.

[3] Hue-Saturation-Value en inglés.

reflejar. Nuevamente, se trata de escoger adecuadamente una base de colores, para que la gama de matices que se pueda representar mezclando componentes de esta base sea lo más extensa posible.

Por ejemplo, si se toma la base: rojo, verde, y azul (RGB). Mezclando las tintas roja (que lo absorbe todo menos el rojo) y verde (que lo absorbe todo menos el verde), se obtiene un color que absorbe toda la radiación y no refleja ninguna, por lo que aparece el color negro. Mezclando rojo y azul o verde y azul ocurre lo mismo. Por ello el rojo, el verde y el azul forman una base que, fuera de los tres colores que posee, no permite obtener muchos más de manera substractiva.

Si la base es celeste, magenta y amarillo (CMY[4]) el número de colores que se pueden obtener de manera substractiva es mayor. Así, por ejemplo, al mezclar un tinte amarillo (que refleja una luz según la teoría triestímulo combinación de rojo y verde) con otro tinte magenta (combinación de azul y rojo) se obtiene un tinte que absorbe el azul y el verde pero refleja el rojo. Así, mientras que la base RGB no permite obtener el color amarillo de manera substractiva, la CMY sí permite obtener el rojo. Este ejemplo ilustra por qué los dispositivos que trabajan con rayos luminosos (como los tubos de rayos catódicos y los dispositivos LCD de los monitores y de las televisiones) eligen la base RGB, mientras que las impresoras, que usan papel y tinta, toman como base la CMY.

1.3. Visión Artificial

La visión artificial tiene como finalidad la extracción de información del mundo físico a partir de imágenes, utilizando para ello un computador. Se trata de un objetivo ambicioso y complejo que actualmente se encuentra en una etapa primitiva.

1.3.1 Representación de la realidad

Un sistema de Visión Artificial actúa sobre una representación de una realidad que le proporciona información sobre brillo, colores, formas, etcétera. Estas representaciones suelen estar en forma de imágenes estáticas, escenas tridimensionales o imágenes en movimiento.

[4] Cyan, magenta, yellow.

Imágenes

Una imagen bidimensional es una función que a cada par de coordenadas (x, y) asocia un valor relativo a alguna propiedad del punto que representa (por ejemplo su brillo o su matiz). Una imagen acromática, sin información de color, en la que a cada punto se le asocia información relativa al brillo, se puede representar como una superficie (ver Figura 16), en la cual la altura de cada punto indica su nivel de brillo. Una imagen en color RGB se puede representar asociando a cada punto una terna de valores que indica la intensidad de tres linternas (una roja, otra verde y otra azul). Una imagen de color de espectro completo se puede representar asociando a cada punto un diagrama espectral de emisión de color.

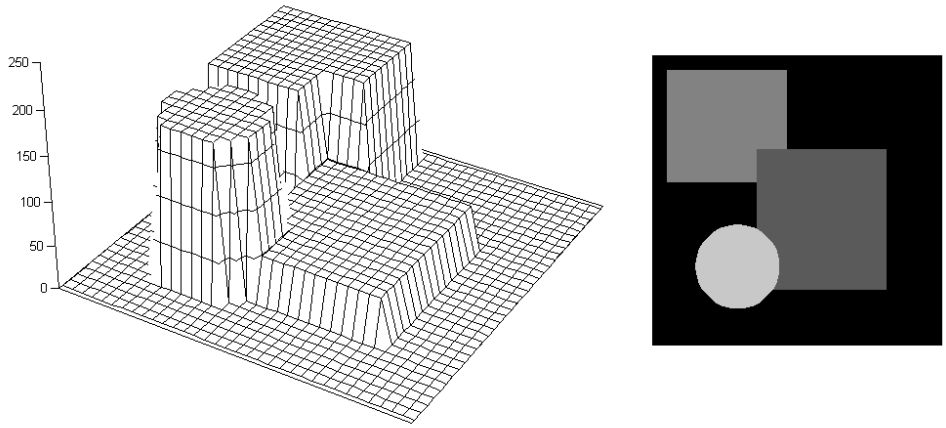


Figura 16.- La imagen plana (2D) de la derecha puede representarse como una superficie, en la que la coordenada z para el punto (x, y) depende del brillo que tenga en la imagen plana.

Escenas 3D

Otra forma de representar la realidad consiste en asignar a cada punto del espacio que pertenece a un objeto (x, y, z) una propiedad del punto (su existencia, su intensidad, su matiz, etcétera.). Al trabajar con imágenes 3D, como se tiene la forma de los objetos, la información de brillo y color puede no ser tan relevante.

Secuencias animadas

Las secuencias de imágenes estáticas dan lugar a secuencias animadas. Por ejemplo, una cámara cinematográfica toma sucesiones de imágenes estáticas que se capturan a una frecuencia determinada. Si estas sucesiones de imágenes se presentan luego a una frecuencia superior a 25 imágenes por segundo aproximadamente, el sistema visual humano no es capaz de distinguir el cambio e interpreta movimiento. Éste es el efecto usado en cine y televisión.

1.3.2 Etapas de un sistema de visión artificial

Se ha visto que el ser humano captura la luz a través de los ojos, y que esta información circula a través del nervio óptico hasta el cerebro donde se procesa. Existen razones para creer que el primer paso de este procesado consiste en encontrar elementos más simples en los que descomponer la imagen (como segmentos y arcos). Después el cerebro interpreta la escena y por último actúa en consecuencia. La visión artificial, en un intento de reproducir este comportamiento, define tradicionalmente cuatro fases principales:

- La primera fase, que es puramente sensorial, consiste en la *captura* o adquisición de las imágenes digitales mediante algún tipo de sensor.

- La segunda etapa consiste en el tratamiento digital de las imágenes, con objeto de facilitar las etapas posteriores. En esta etapa de *procesamiento previo* es donde, mediante filtros y transformaciones geométricas, se eliminan partes indeseables de la imagen o se realzan partes interesantes de la misma.

- La siguiente fase se conoce como *segmentación*, y consiste en aislar los elementos que interesan de una escena para comprenderla.

- Por último se llega a la etapa de *reconocimiento* o *clasificación*. En ella se pretende distinguir los objetos segmentados, gracias al análisis de ciertas características que se establecen previamente para diferenciarlos.

Estas cuatro fases no se siguen siempre de manera secuencial, sino que en ocasiones deben realimentarse hacia atrás. Así, es normal volver a la etapa de segmentación si falla la etapa de reconocimiento, o a la de preproceso, o incluso a la de captura, cuando falla alguna de las siguientes.

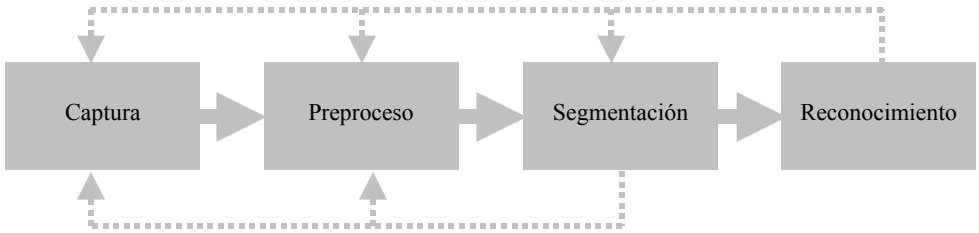


Figura 17.- Diagrama de bloques de las etapas típicas en un sistema de visión artificial.

1.3.3 Configuración informática de un sistema de visión artificial

Aunque se pueden proponer configuraciones muy avanzadas, por ejemplo incluyendo hardware específico para acelerar ciertas operaciones, los elementos imprescindibles son:

- Un *sensor óptico* para captar la imagen: Una cámara de vídeo, una cámara fotográfica, una cámara digital, un escáner... uniéndole un conversor analógico-digital cuando sea preciso.

- Un *computador* que almacene las imágenes y que ejecute los algoritmos de preprocesado, segmentación y reconocimiento de la misma.

1.4. Bibliografía del capítulo

[GW93] caps. 1 y 2.

[Bax94] caps. 1 y 2.

[F+97] cap. 13.

[Cas 85] caps. 32, 33 y 35.

Capítulo 2 Adquisición y representación de imágenes digitales

En este capítulo se tratan los aspectos más relevantes del proceso de *captura* y *digitalización* de una imagen, esto es, la adquisición de la imagen del mundo físico y su paso al dominio discreto y virtual informático.

Una vez digitalizada una imagen bidimensional queda constituida por un conjunto de elementos llamados *píxeles*[5]. Cada píxel ofrece cierta información sobre una región elemental de la imagen. En imágenes en blanco y negro esta información es el brillo. En imágenes en color, la información corresponde a la intensidad de cada una de las componentes de una base de color (por ejemplo RGB). Dentro de este capítulo también se repasan las técnicas de compresión, que buscan la forma más eficiente de almacenar las imágenes digitales.

Se finaliza el capítulo con el estudio de las relaciones básicas que se pueden establecer entre los píxeles de una imagen.

---

[5] Del inglés pixel que abrevia a "picture element".

2.1. Captura y digitalización de imágenes

Las imágenes digitales son "señales" discretas, que suelen tener origen en una "señal" continua. Por ejemplo, una cámara digital toma imágenes del mundo real que es continuo (tanto el espacio, como el espectro emitido por los objetos se consideran continuos); otro ejemplo es el de un escáner, el cual digitaliza imágenes procedentes de documentos o fotografías que a efectos prácticos también se consideran continuos.

En el proceso de obtención de imágenes digitales se distinguen dos etapas. La primera, conocida como *captura*, utiliza un dispositivo, generalmente óptico, con el que obtiene información relativa a una escena. En la segunda etapa, que se conoce como *digitalización*, se transforma esta información, que es una señal con una o varias componentes continuas, en la imagen digital, que es una señal con todas sus componentes discretas.

2.1.1 Modelos de captura de imágenes

A grandes rasgos, para capturar una imagen se suele distinguir entre dispositivos pasivos (basados generalmente en el principio de *cámara oscura*) y dispositivos activos (basados en el *escaneo*), aunque también se pueden realizar mediciones de propiedades sobre un objeto y posteriormente construir una imagen de manera sintética.

El modelo de lente fina

Desde hace mucho tiempo es conocida la manera de formar una imagen utilizando el principio de cámara oscura. Este dispositivo pasivo está constituido por una caja cerrada, conocida como cámara, en una de cuyas paredes existe un orificio para el paso de la luz. La luz, tras entrar en la cámara, se proyecta sobre la pared opuesta a la que tiene el orificio, obteniéndose allí una imagen invertida de lo que está fuera de la cámara. Cardan, en 1550, tuvo la idea de colocar una lente delante de dicho orificio para aumentar la luminosidad. Para explicar el funcionamiento del conjunto se usa *el modelo de lente fina*.

El modelo de lente fina explica que una lente de grosor despreciable y perfectamente biconvexa permite recoger la luz de una escena y proyectarla de manera nítida sobre una superficie llamada *plano de formación de la imagen*. Este comportamiento es posible gracias a las propiedades que tiene tal lente:

- Es una lente biconvexa de grosor despreciable cuya sección posee dos ejes de simetría. Al mayor se le denomina *eje axial*, al menor *eje óptico*. El *centro óptico* es el punto interior a la lente donde se cortan ambos ejes.

- Todo haz de luz que pasa por el centro óptico de una lente continúa en línea recta (Figura 18 a).

- Todos los haces paralelos que inciden perpendiculares al eje axial de una lente, tras atravesarla, se cortan en un punto llamado *foco* (Figura 18 a) situado sobre el eje óptico. Se define *distancia focal* como la distancia del foco al centro óptico de la imagen.

- Sea un punto *P*, que se encuentra a una distancia de la lente mucho mayor a la distancia focal. Todos los rayos que provengan de *P*, tras atravesar la lente, se cortan en un punto llamado *punto de formación de la imagen* (Figura 18 b).

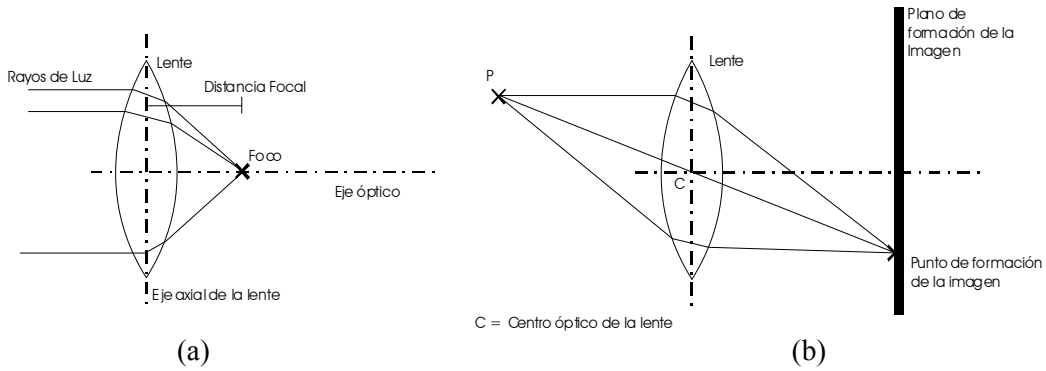


Figura 18.- Trayectoria seguida por la luz al atravesar una lente biconvexa de grosor despreciable. (a) los haces paralelos que inciden perpendiculares al eje de la lente se cortan en el foco. (b) los haces provenientes de un mismo punto objeto se cortan en el punto de formación de la imagen.

De esta última propiedad se deduce que los puntos de formación de la imagen correspondientes a puntos *P* que están a la misma distancia de la lente, forman un plano perpendicular al eje óptico de la lente, que es el plano de formación de la imagen. Así, a una figura formada por un conjunto de puntos *P* que equidistan de la lente se corresponde una figura semejante, aunque invertida, en el plano de formación de la imagen.

La distancia del plano de formación de la imagen al eje axial de la lente está relacionada con la distancia del punto *P* al mismo y con la distancia focal de la lente. Si llamamos S_0 a la distancia del punto *P* a la lente, S_i a la distancia de la lente al plano de formación de la imagen para ese punto *P* y *f* a la distancia focal, se cumple la relación:

$$\frac{1}{S_i} - \frac{1}{S_0} = \frac{1}{f}$$

Siendo S_i/S_0 la relación de aumento entre la imagen real y la imagen proyectada.

Además, sobre la Figura 18 se aprecia que la variación de la distancia del plano de formación de la imagen, respecto de la lente, se puede utilizar para concentrar más o menos los haces procedentes del punto *P* sobre tal plano, proceso conocido como *enfoque*.

Las lentes reales se construyen utilizando un material transparente llamado vidrio óptico, mezcla de productos químicos como el oxido de bario, lantano y tántalo. Las lentes se diseñan con una geometría tal que se obtengan los resultados descritos, utilizando fundamentalmente una propiedad de la luz que consiste en su cambio de dirección al pasar de un medio a otro. El ángulo de este cambio depende del medio que se atraviese y del ángulo del rayo de luz con la normal a la superficie que separa los dos medios. El *índice de refracción* mide la dependencia de este cambio respecto al medio atravesado. Por ejemplo, el vacío tiene índice de refracción 1'0, el agua 1'333, el vidrio normal 1'528 y el vidrio óptico 1'519.

Una lente como la descrita en el modelo de lente fina no puede conseguirse en la realidad. Por ejemplo, las lentes reales tienen un grosor que no es despreciable y esto provoca aberraciones cromáticas[6].

[6] La aberración cromática se debe a que el índice de refracción es diferente según la longitud de onda de la luz que atraviesa la lente. Así, diferentes colores dan lugar a diferentes planos de formación de la imagen, y esto da lugar a la aparición de bandas de colores en los bordes de los objetos dentro de una imagen.

Otra diferencia con el modelo de lente fina radica en que el grosor de la lente varía a lo largo de la lente y esto crea aberraciones esféricas[7]. Otros aspectos que desvían su comportamiento del ideal son la existencia de defectos en el cristal o la no ausencia total de color en el mismo.

Es por todo esto que las propiedades descritas sólo se cumplen de manera aproximada en la realidad, aunque en general, cuanto mejor sea la calidad de una lente más se aproximará su comportamiento al ideal.

La cámara oscura

El principio de cámara oscura, descrito en el punto anterior, se puede usar para capturar imágenes de escenas tridimensionales (del mundo real) y proyectarlas en un plano bidimensional. Dispositivos de este tipo son las cámaras fotográficas y las cámaras de vídeo. Este modelo además se puede usar para capturar imágenes de elementos bidimensionales, como fotografías y documentos, como por ejemplo hacen los escáneres de cámara. También se pueden usar dos o más cámaras para capturar diferentes perspectivas de una misma escena y construir una representación 3D de la misma.

El escaneo

Este esquema es fundamentalmente distinto del basado en cámara, ya que existe un elemento activo (generalmente un haz de luz láser) que recorre la escena que se desea capturar. Por tanto son imprescindibles dos dispositivos, uno emisor del haz de luz y otro el receptor. El escáner emite el haz de luz y éste, tras chocar con la imagen que se escanea, es recogido en el detector de luz. Repitiendo este proceso de manera continua se puede construir una señal que corresponde a una representación de la escena.

Los dispositivos basados en el escaneo también se usan con diferentes fines. Así, los escáneres-laser pueden capturar escenas 3D directamente, y los escáneres de tambor permiten capturar imágenes de elementos bidimensionales.

[7] La distorsión esférica se origina al existir diferente plano de formación de la imagen para los rayos que atraviesan la zona más gruesa de la lente que para los que atraviesan la zona más delgada de la misma.

Los dispositivos basados en cámara aventajan a los basados en escaneo en velocidad. Además son más simples y se parecen más al sistema visual humano. Es de prever que, con el tiempo, los modelos de cámara terminen superando también a los de escaneo en cuanto a calidad de la imagen obtenida, ya que su principal cuello de botella, que se encuentra actualmente en el elemento digitalizador, parece que puede ser mejorado sensiblemente.

2.1.2 La digitalización

Es el proceso de paso del mundo continuo (o analógico) al mundo discreto (o digital). En la digitalización normalmente se distinguen dos procesos: el *muestreo* ("sampling") y la *cuantización* ("quantization").

Muestreo

El muestreo de una señal continua consiste en la medición a intervalos (discretización) respecto de alguna variable (generalmente el tiempo o el espacio), siendo su parámetro fundamental la *frecuencia de muestreo*, que representa el número de veces que se mide un valor analógico por unidad de cambio. Mediante el muestreo se convierte una imagen I_C, que es algo continuo, en una matriz discreta I_D de $N \times M$ píxeles. El número de muestras por unidad de espacio sobre el objeto original conduce al concepto de *resolución espacial* de la imagen. Ésta se define como la distancia, sobre el objeto original, entre dos píxeles adyacentes. Sin embargo la unidad de medida de resolución espacial mas habitual suele ser los píxeles por pulgada (comúnmente DPIs[8]) siempre medidos sobre el objeto original.

De esta forma, el proceso de muestreo, para una imagen, que asocia a cada punto un valor real, cambia una imagen del formato:

$$I_C(x, y) \in \Re \text{ en donde } x, y \in \Re$$

al formato:

$$I_D(x, y) \in \Re \text{ en donde } x, y \in \mathbb{N} \text{ y } 0 \leq x \leq N\text{-}1 \text{ , } 0 \leq y \leq M\text{-}1$$

[8]Dots per inch en inglés.

que se puede representar en forma matricial:

$$I_D(x,y) = \begin{pmatrix} I_D(0,0) & I_D(0,1) & \dots & I_D(0,M-1) \\ I_D(1,0) & I_D(1,1) & \dots & I_D(1,M-1) \\ \dots & & & \dots \\ I_D(N-1,0) & I_D(N-1,1) & \dots & I_D(N-1,M-1) \end{pmatrix}$$

Cuantización

La segunda operación es la cuantización de la señal, que consiste en la discretización de los posibles valores de cada píxel. Los niveles de cuantización suelen ser potencias de 2 para facilitar su almacenamiento en el computador. Así, suelen usarse 2, 4, 16 ó 256 niveles posibles. De esta forma, I_D que pertenece a $\Re$ se convierte en I_{DC} (discreta cuantizada) que pertenece a $\mathbb{N}$. El número de niveles posibles define la *resolución radiométrica*.

$$I_{DC}(x,y) \in \mathbb{N}$$

Donde,

$$x, y \in \mathbb{N} \text{ y } 0 \le x \le N\text{-}1 \text{ , } 0 \le y \le M\text{-}1$$

$$0 \le I_{DC}(x,y) \le 2^q\text{-}1$$

Cuando las imágenes sólo tienen información sobre el brillo se habla de *imágenes en niveles de gris* y se suelen utilizar hasta 256 niveles para representar los tonos intermedios desde el negro (0) hasta el blanco (255). Si sólo se permiten dos niveles de cuantización (normalmente blanco y negro) se habla de *imágenes bitonales* o *imágenes binarias*. Para el caso del color suelen usarse 256 niveles para representar la intensidad de cada uno de los tres colores primarios (RGB). De esta forma se obtienen 16 millones de colores aproximadamente (256x256x256) y se habla de *imágenes en color real*. En algunos casos puede necesitarse mayor resolución radiométrica y se usan 4096 niveles por banda de color en vez de 256, o incluso más.

Elección de las resoluciones espaciales y radiométricas

El proceso de digitalización requiere evaluar qué resolución espacial y qué resolución radiométrica se precisan para representar adecuadamente una imagen. Dicho de otra forma, con qué frecuencia se muestrean los píxeles (frecuencia de muestreo), y qué gama de colores se permite (elección de la paleta).

El *teorema de muestreo* de *Nyquist* establece la frecuencia mínima que es preciso aplicar para poder recuperar todos los elementos que se repiten con una frecuencia máxima en una señal. El teorema establece que la frecuencia de muestreo debe ser igual al doble de la máxima frecuencia con que se puedan repetir los elementos que se quieran capturar en la señal.

Sobre una imagen a impresa a 200 DPIs la frecuencia máxima con la que pueden cambiar los pixeles es de 100 cambios por pulgada. Para obtener toda la información posible al digitalizar tal imagen es preciso muestrear, al menos, a la misma resolución. Sin embargo suele aplicarse un margen doble de seguridad en el muestreo, según el cual la frecuencia de muestreo debe tomarse igual al doble de la resolución original, y por ello en este caso se digitalizaría a 400 DPIs.

También hay que tener en cuenta que dependiendo del uso que se vaya a hacer de una imagen, la elección de los parámetros de digitalización puede variar de una forma menos objetiva. Así, para la publicación de un periódico en blanco y negro, 16 niveles de intensidad podrían ser suficientes, pero elegir menos de 80 por 80 píxeles por pulgada de resolución espacial sería inadmisible; mientras que para una imagen, con vistas a su reconocimiento, aunque podría ser preciso utilizar más niveles de intensidad, se podría permitir una resolución espacial menor.

Para ilustrar estos aspectos en la Figura 19 se presenta la imagen de Lena[9] digitalizada con diferentes resoluciones espaciales y radiométricas.

[9] La imagen de Lena es una imagen clásica dentro del mundo del procesado digital de imágenes. Es una imagen de una chica, aparecida en la publicación Play Boy en 1972, escaneada por un investigador desconocido. La ventaja de operar sobre imágenes estándar radica en que permiten comparar los resultados que se obtienen con los que han obtenido otros investigadores.

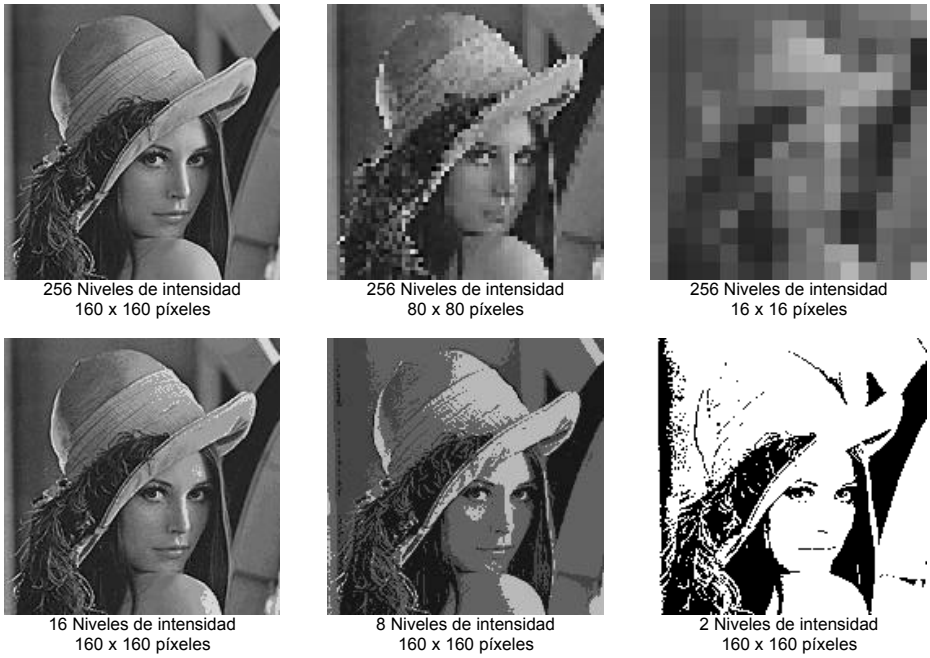


Figura 19.- En la fila superior se presenta la misma imagen, siempre a 256 niveles de intensidad, usando diferentes resoluciones espaciales. En la fila inferior se mantiene la resolución espacial y se reduce el nivel de cuantización.

En la Figura 20b se presenta otro ejemplo, en él se aprecia cómo una reducción de la resolución espacial de la imagen, conseguida dejando uno de cada 4 píxeles, produce una pérdida de la legibilidad del documento binario de la Figura 20a. Dicha pérdida no sería tan patente si se hubiese usado un método de reescalado más adecuado, como por ejemplo el que consiste en construir una imagen que interpola los valores de cada grupo de 4 píxeles aumentado la resolución radiométrica para representarlos (ver Figura 20c). Este método de reescalado implica un intercambio no reversible de valores entre la resolución espacial (que disminuye) y la radiométrica (que aumenta).

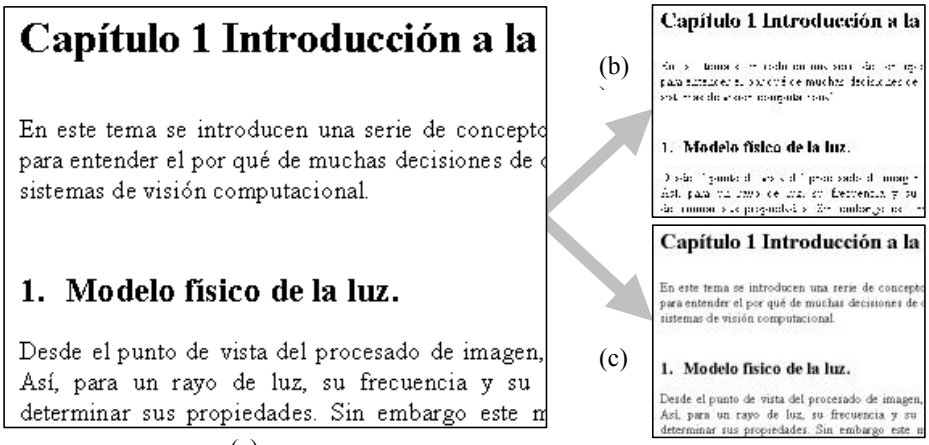


Figura 20.- Efecto de la reducción de resolución sobre una imagen. La imagen (a) corresponde a un texto y se ha tomado con un escáner bitonal; (b) la misma imagen tras reducir su resolución en un 50% respecto de la original cuando simplemente se deja uno de cada cuatro píxeles; (c) la misma imagen tras reducir su resolución en un 50% utilizando un promediado de cada grupo de cuatro píxeles.

Muestreo y cuantificación no uniformes

Hasta ahora se ha tratado el concepto de muestreo y cuatización como si fuese un proceso uniforme. El *muestreo no uniforme* consiste en el uso de diferente frecuencia de muestreo para diferentes zonas de la imagen. De esta forma, las zonas más interesantes pueden tener una resolución espacial mayor que las menos interesantes, consiguiendo un ahorro de los recursos del sistema.

La *cuantización no uniforme* se basa en el uso de *paletas*. Una paleta consiste en un conjunto de colores a los que se les asigna una referencia. Los píxeles de las imágenes que usan paletas contienen como valor la referencia al color de la paleta que quieren presentar. Cuando el número de colores de una imagen es pequeño, el uso de paletas permite, además de un ahorro de memoria, simplificar ciertas operaciones, como el cambio de un color por otro dentro de una imagen, que sólo exige el cambio de la paleta y no el cambio directo de todos los píxeles de la imagen.

Ejemplo 2.-

Para saber cuántos bytes ocupa una imagen de 640x480 píxeles con 256 niveles de intensidad cuando se representa en una pantalla de ordenador, se opera:

$$\text{Nº de píxeles} = 640 \times 480 = 307.200 \text{ (píxeles)}$$

Como se utilizan 256 niveles de intensidad para codificar 256 valores se necesita 1 byte por píxel, así:

$$\text{Nº de bytes} = 307.200 \times 1 \approx 300 \text{ (Kb)}$$

Ejemplo 3.-

Para saber cuantos bytes ocupa una imagen de 1024 por 768 píxeles con codificación para 16 millones de colores (color Real) se opera:

$$\text{Nº de píxeles} = 1024 \times 768 = 786.432 \text{ (píxeles)}$$

En este caso cada píxel necesita 3 bytes (uno para codificar 256 niveles de rojo, otro para 256 niveles de azul, y otro para 256 de verde), por tanto:

$$\text{Nº de bytes} = 786.432 \times 3 = 2.359.296 \text{ (bytes)} \approx 2'3 \text{ (Mb)}$$

2.1.3 Dispositivos de captura

En los siguientes apartados se analizan los principales dispositivos (fundamentalmente *cámaras* y *escáners*) que se pueden encontrar en el mercado para realizar procesos de captura. En la Figura 21 se muestra cómo se relacionan estos dispositivos con un computador.

Cámara fotográfica analógica

Básicamente, una cámara fotográfica está constituida por un recinto oscuro (la cámara), en la que se ha montado un *objetivo*. El objetivo está formado por un conjunto de lentes que tienen la misión de comportarse como una única lente que seguiría el modelo ideal de lente fina, intentando corregir las aberraciones que se producen al utilizar lentes reales.

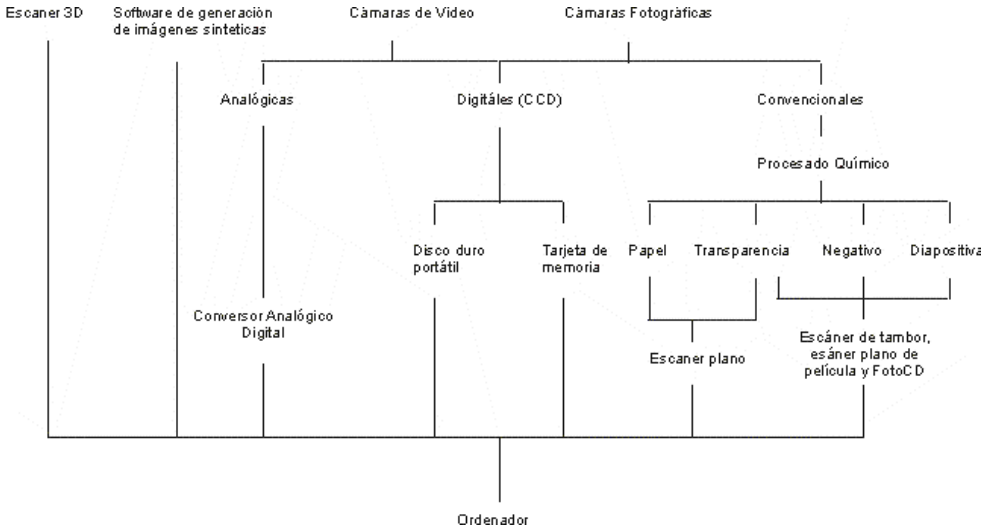


Figura 21.- Cuadro de relación entre dispositivos y el computador.

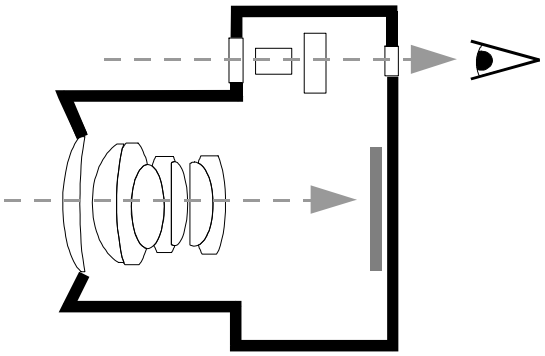


Figura 22.- Esquema de una cámara analógica de fotografía.

El objetivo forma la imagen luminosa en el interior de la cámara, en el plano de formación de la imagen, donde hay una superficie sensible a la luz. Entre el objetivo y la superficie sensible se encuentra el *obturador*, que sólo deja pasar la luz en el momento de captura de la imagen. Este momento lo determina el fotógrafo presionando el disparador. Para permitir al usuario encuadrar el objeto el aparato dispone de un visor. Por último, para obtener la imagen, es necesario realizar un proceso químico sobre la superficie sensible, que se conoce como *revelado*. Este proceso no es reversible, por lo que la superficie sensible es de un solo uso.

Normalmente las cámaras fotográficas permiten variar la distancia a la que se encuentra la lente ideal del plano de formación de la imagen (donde está la película). Este proceso permite el enfoque, es decir el ajuste de la definición de la imagen de los objetos que se encuentren a una distancia determinada de la cámara.

Las cámaras suelen permitir también variar la cantidad de luz que entra en la misma mediante un dispositivo conocido como *diafragma*. Cuando se abre mucho el diafragma, entran muchos rayos de luz por cada punto *P* de la escena, y de acuerdo con los principios que se enunciaron para el modelo de lente fina, esto hace que sólo los elementos que estén a cierta distancia de la cámara aparezcan enfocados. Por el contrario, cuanta menor es la apertura del diafragma menos rayos de luz entran por cada punto *P* de la escena. En el límite, cuando por cada punto *P* de la escena sólo incidiese un rayo en el plano de formación de la imagen, toda la imagen debería aparecer enfocada simultáneamente. Así, en el caso de poca apertura se dice que se tiene gran *profundidad de campo*, y en el caso de mucha apertura se dice que se tiene poca amplitud de campo.

Desgraciadamente, cuando la apertura es pequeña el tiempo que debe dejarse abierto el obturador (*tiempo de exposición*) debe ser grande, pues en otro caso no entra suficiente luz como para impresionar la película. Aunque es posible aumentar la sensibilidad de la película, para que con menos luz quede impresionada, existe la contrapartida de que a mayor sensibilidad de la película menor definición posee la misma (el *grano* de la película es más grueso[10]).

Por último se debe señalar que existen objetivos con diferente distancias focales. Para cada distancia focal se obtiene diferente tamaño en la representación de un objeto (ampliación). Hay también objetivos de focal variable que permiten cambiar la distancia focal dentro de un rango de valores (zoom). La problemática que introducen estos objetivos es más compleja. En ellos, por ejemplo, las aberraciones son más difíciles de corregir. Por ello sólo son útiles para aquellos problemas en los que la calidad de la imagen no sea un factor muy importante.

[10] Se llama grano a cada partícula de haluro de plata, sustancia reactiva a la luz que incide sobre ella utilizada en las películas fotográficas. Cuanto mayor es el grano mayor sensibilidad a la intensidad se consigue, pero menor definición y detalle tiene la imagen.

Cámara de vídeo analógica

La cámara de vídeo es un aparato que transforma una secuencia de escenas ópticas en señales eléctricas. Está constituida por un objetivo, un tubo de cámara y diversos dispositivos electrónicos de control. La luz se enfoca dentro del tubo de cámara sobre una superficie fotosensible que convierte la señal lumínica en una señal eléctrica denominada *señal de vídeo*. Esta señal consiste en una onda en la cual la intensidad de cada punto de cada línea de la pantalla se describe por la amplitud de la onda. La onda contiene la información de cada línea de la pantalla separada por una señal de control, y a su vez, cada imagen que está separada de la siguiente por otra señal de control (ver Figura 23).

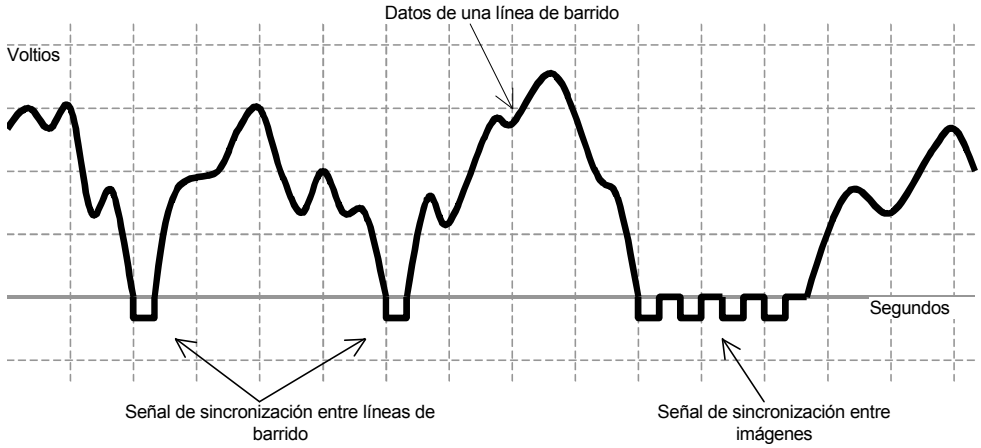


Figura 23.- Representación de una señal de video analógico.

La conversión de la señal lumínica en señal eléctrica se realiza sobre una superficie fotosensible, llamada *diana*, cuyo diámetro oscila entre 12 y 30 mm. Cuanto mayor sea esta superficie mayor resolución se puede obtener. En las cámaras domésticas analógicas se utiliza un tipo de diana fotoconductora llamada *vidicon* de unos 17 mm de diámetro.

Finalmente, para obtener una imagen digital se precisa una *tarjeta digitalizadora de vídeo*. La calidad de este dispositivo depende del número de muestras que es capaz de tomar de la señal de vídeo por unidad de tiempo, y de la resolución radiométrica que es capaz de alcanzar.

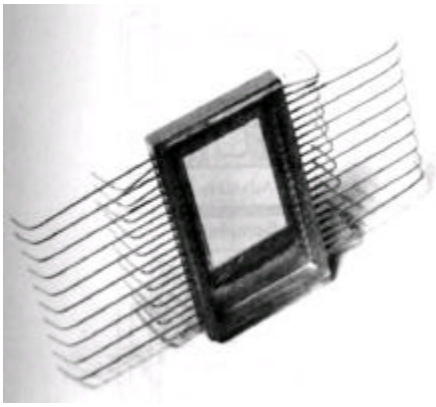


Figura 24.- Primer CCD comercial. Constaba de 120.000 elementos y tenía un tamaño de 0’5x0’25 pulgadas.

Cámara digital de fotografía y vídeo

El esquema de ambas cámaras es idéntico al de sus correspondientes analógicas, con la diferencia de que el dispositivo sensible es un *CCD*[11] (ver Figura 24). Un CCD es un dispositivo constituido por una matriz de elementos fotosensibles, que se sitúa en el mismo lugar que el plano de formación de la imagen, de manera que se forma la imagen sobre él.

El número de elementos fotosensibles, junto con el área que ocupan, definen la resolución espacial del dispositivo.

Por otro lado, cada uno de estos elementos fotosensibles es capaz de obtener una carga eléctrica proporcional a la intensidad de la luz que le incide. Después la carga eléctrica de cada celda se transmite a un amplificador eléctrico. El tiempo que tarda esta operación determina el número de imágenes por segundo que puede tomar el dispositivo.

Así, desde el punto de vista de la resolución espacial el CCD es un dispositivo digital, mientras que desde el de la resolución radiométrica puede considerarse analógico. Aunque finalmente, esta información de carga analógica

[11]Dispositivo de Carga Acoplada (“Coupled Charge Device”).

puede ser discretizada mediante un conversor analógico digital, fijandose en este punto de manera digital la resolución radiométrica del CCD.

Originalmente los dispositivos CCD registraban únicamente la intensidad de luz incidente sobre cada celda. La solución más sencilla para conseguir una imagen en color consiste en cubrir la retícula de celdas con filtros que sólo permitan el paso de cada una de las componentes RGB. Estos filtros se muestrean en grupos de cuatro (RGBG) para aportar la información del color de un píxel. Se disponen dos filtros para el verde para solventar la geometría difícil del 3 y para emular la mayor sensibilidad del ojo hacia el verde. Un problema que aparece con esta técnica consiste en que los cuatro receptores que aportan la información hacia un mismo píxel no ocupan la misma posición física. Por ello, en los bordes de los objetos que aparecen en las imágenes pueden aparecer distorsiones del color.

Para evitar estos problemas podrían utilizarse 3 CCDs (uno por cada plano de color), aunque esta solución es más cara y voluminosa. En el año 2002 ha aparecido un nuevo tipo de dispositivos (denominados *Foveon*) que incorpora los tres receptores RGB en la misma posición física mediante un sistema multicapa.

Escáner de cámara

Este dispositivo, que sigue el modelo de cámara, recorre una imagen plana (un documento, una fotografía, un plano...) con un CCD compuesto por una única línea de elementos fotosensibles, llamado CCD lineal. En su recorrido, el CCD lineal construye una representación digital de la imagen.

Se pueden distinguir dos tipos de escáneres de cámara: los fijos, que mueven el haz de luz para recorrer el documento, y los de rodillo, que mantienen fijo el haz de luz y mueven el documento a escanear. Los de rodillo tienen su principal atractivo en la reducción de espacio que ocupa el dispositivo, y en la facilidad para la alimentación automática de documentos. Los fijos permiten un ajuste más exacto del papel (que no se mueve).

Una propiedad interesante de estos dispositivos es que mientras la resolución en una de las dimensiones viene determinada por el número de celdas receptoras en el CCD lineal, la resolución en la dimensión perpendicular depende de la velocidad relativa a la que se desplace respecto al elemento escaneado.

Escáner de tambor

Este tipo de escáner se utiliza para digitalizar elementos planos (documentos, fotografías, etc.). El elemento que se desea escanear se sitúa sobre un cilindro denominado tambor. Allí, se escanea usando un dispositivo que emite un haz puntual sobre el mismo. Este haz, tras reflejarse en el elemento que se escanea se recoge en un detector sensible. Tras analizar el haz recibido se construye una representación del elemento escaneado.

Escáner 3D o sensor de rango

Los *sensores de rango* se utilizan para reconstruir la estructura 3D de una escena. Capturan imágenes en las que está codificada la forma 3D de los objetos midiendo la profundidad de sus superficies. Son apropiados en aplicaciones que requieren medir distancias 3D (por ejemplo para desviar objetos móviles de obstáculos) o para estimar la forma de la superficie de un objeto (por ejemplo en la inspección de objetos en industrias).

Si para muestrear la superficie de un objeto un elemento móvil la recorre tocándola, se denominan sensores de contacto (o táctiles). Estos a su vez pueden ser manuales o automáticos. Generalmente consisten en un brazo con varias articulaciones que está fijado a un soporte. En el extremo del brazo hay un puntero que se rastrea sobre la superficie del objeto. Son más baratos que los digitalizadores no táctiles.

Los sensores de rango no táctiles se clasifican en activos o pasivos. Son sensores de rango activos los que o bien proyectan haces controlados de energía (luz o sonido sobre la escena) desde una posición y orientación conocidas, o bien analizan el efecto de cambios controlados en algún parámetro del sensor (por ejemplo el foco). Los primeros detectan la posición del haz en el objeto para realizar una medida de la distancia. Los sensores de rango activos pueden utilizar una gran variedad de principios físicos: radares, sonoros, interferometría holográfica, enfoque, triangulación activa... Si utilizan fenómenos ópticos para adquirir las imágenes de rango, se denominan también sensores de rango ópticos.

Un ejemplo de sensor de rango óptico es el *escáner 3D láser*. Este dispositivo obtiene gran densidad de puntos de forma precisa pero es mucho más caro que un digitalizador táctil. Para obtener los puntos de la superficie de un objeto utiliza un método conocido como *tiempo de vuelo*, que básicamente consiste en medir el tiempo que tarda en recibirse el rebote del elemento que escanea y que

se conoce como *vóxel*[12]. Si el objeto a percibir es oscuro, el láser no es reflejado, por lo que las zonas negras o muy oscuras pueden no aparecer. Actualmente existen digitalizadores 3D de tamaño pequeño, de fácil uso, que realizan un muestreo en menos de un segundo, generando hasta 200x200 puntos 3D, información de color en cada punto, e información de la conectividad de los mismos, proporcionando una representación de superficie en forma de mallado poligonal (ver Figura 25).

Los sensores pasivos son los no considerados como activos: se basan únicamente en imágenes de niveles de gris o color para reconstruir la profundidad y son los usados para visión estéreo como se verá en el capítulo 6.

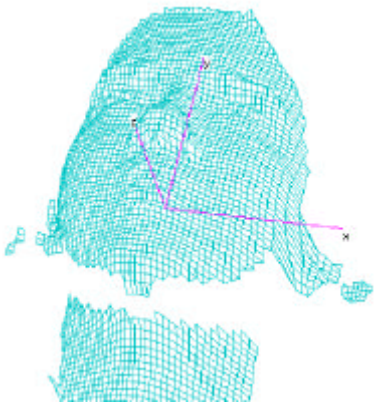


Figura 25.- Imagen obtenida con un escáner 3-D láser.

2.2. Representación de la imagen y estructuras de datos

Las imágenes suelen almacenarse en los ordenadores en forma de ficheros. En este punto se analizarán las estructuras que se usan a tal efecto, los métodos utilizados para optimizar el espacio requerido y algunos de los diferentes formatos estándar (TIFF, GIF, BMP, JPG...).

[12] Del inglés voxel que juega con la abreviatura de "volume element" y con el parecido a la palabra píxel.

2.2.1 Estructura del fichero de imagen

Generalmente una imagen almacenada en un ordenador está constituida (ver Figura 26) por un mapa de bits (sería mejor decir de píxeles) precedido por una cabecera que describe sus características (tamaño de la imagen, modo de color, paleta, resolución de la imagen...). Frecuentemente, cuando la imagen se encuentra en la memoria principal del ordenador la cabecera y el mapa de bits no están contiguos.

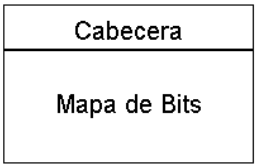


Figura 26.- Esquema de un fichero gráfico.

Ejemplo 4.-

Un formato de fichero muy sencillo para imágenes en niveles de gris podría constar de:

- Una cabecera, donde se indicaría el tamaño de la imagen mediante dos números enteros *N* y *M*.

- Un mapa de bits con *N*x*M* números, en formato ASCII y separados por espacios. Utilizando el 0 para indicar el color negro, el 255 para indicar el color blanco, y los números intermedios para intensidades entre blanco y negro.

Este ejemplo corresponde al formato gráfico comercial PGM.

2.2.2 Compresión de imágenes

En ocasiones es impracticable el tratamiento directo de ciertas imágenes debido a la gran cantidad de datos que requiere su almacenamiento o su transmisión. La *compresión* de las imágenes trata este problema, mediante la reducción de la cantidad de datos necesarios para representar una imagen digital.

El primer punto que se debe tener en cuenta es que no son lo mismo "los datos usados para presentar una imagen" que "la información que contienen". Por ejemplo: la idea del número "dos" se puede representar como: 2, dos, II, 6-3-1... Cada representación necesita diferente espacio para su codificación (1, 3, 2 y 5 caracteres respectivamente) pero codifican la misma información.

Si n_1 y n_2 denotan el tamaño de los datos necesarios para representar la misma información en dos sistemas diferentes s_1 y s_2, definimos la *razón de compresión* C_R de n_1 frente a n_2 como:

$$C_R = \frac{n_1}{n_2}$$

Se define la *redundancia relativa* R_D como:

$$R_D = 1 - \frac{1}{C_R}$$

Estudiando estas fórmulas se observa que:

Si $n_1 = n_2$ Þ $C_R = 1$ $\Rightarrow R_D = 0$ (no hay redundancia de n_1 respecto a n_2)

Si $n_1 >> n_2$ Þ C_R ® ∞ $\Rightarrow R_D$ ® 1 (n_1 es muy redundante respecto a n_2)

Si $n_1 << n_2$ Þ C_R ® 0 $\Rightarrow R_D$ ® -∞ (n_2 es muy redundante respecto a n_1)

Así, se dice que un código es más redundante que otro cuando precisa más datos que aquél para describir la misma información. Clásicamente se distinguen tres tipos de redundancia:

- Redundancia en la codificación
- Redundancia en la representación espacial de los píxeles
- Redundancia visual

Redundancia en la codificación

Hasta ahora se ha usado un tamaño fijo para representar la información de cada punto de las imágenes. Por ejemplo, se ha usado un *byte*[13] para representar la intensidad de cada punto de una imagen con un nivel de cuantización de 256 niveles de gris.

Sin embargo en una imagen suelen existir niveles de intensidad que son más probables que otros porque aparecen más veces. La codificación de tamaño variable es una técnica de compresión que aprovecha esta circunstancia. Consiste en asignarle un código más corto a los niveles de intensidad más probables (que aparecen más veces) y más largo a los menos probables (que aparecen menos veces), consiguiendo una reducción del tamaño de los datos de la imagen. Un ejemplo de este tipo de compresión lo define el método de *compresión Huffman*.

Ejemplo 5.-

Una imagen de 8 niveles de gris se codifica según la Tabla 1. Para calcular el nivel de redundancia de la codificación de tamaño fijo (que usa tres bits por píxel y tiene 50x50 píxeles de tamaño) respecto de la que usa compresión mediante codificación de tamaño variable se debe calcular la cantidad de memoria necesaria para cada representación. Con el código original, para representar la imagen hacen falta:

$$50x50x3 \text{ bits} = 7500 \text{ bits}$$

Con el código de tamaño variable se tiene que para cada píxel, en media, se usan:

$$2x(0'19)+2x(0'25)+2x(0'21)+3x(0'16)+4x(0'08)+5x(0'06)+6x(0'02) = 2.7 \text{ bits}$$

Así, para representar la imagen se necesitarían:

$$50x50x2'7 = 6750 \text{ bits}$$

Obteniendo unos valores para C_R y R_D:

[13] Un byte está compuesto de 8 bits. Un bit es la unidad mínima de información en un computador.

$$C_R = \frac{7500}{6750} = 1.\hat{1}$$

$$R_D = 1 - \frac{1}{1.\hat{1}} = 1 - 0.0\hat{9} = 0.1$$

Es decir, que según esta codificación de tamaño variable, aproximadamente el 10% del tamaño de los datos representados por la codificación de tamaño fijo son redundantes.

Sin compresión	Probabilidad de aparición	Código de tamaño variable
000	0.19	11
001	0.25	01
010	0.21	10
011	0.16	001
100	0.08	0001
101	0.06	00001
110	0.03	000001
111	0.02	000000

Tabla 1.- Ejemplo de codificación de tamaño variable.

Redundancia en la representación espacial de los píxeles

Una figura regular ofrece una alta correlación entre sus píxeles. Esta correlación da lugar a una redundancia espacial o geométrica si la forma de representación no es la adecuada. La forma más usual de tratar esta redundancia es mediante el empleo de rachas (*runs*) en la codificación.

La codificación mediante rachas indica cada vez el próximo elemento de la imagen y cuánto se repite. Si se aplica este método a imágenes en blanco y negro, se puede omitir cuál es el próximo elemento ya que siempre será distinto al

anterior (blanco-negro-blanco...), tomando la convención de empezar siempre, por ejemplo, por blanco.

La compresión mediante rachas se usa en el estándar *CCITT*[14] *Grupo 3* combinada con la compresión Huffman. Existe una ampliación bidimensional de la idea de compresión por rachas, que se usa en el estándar *CCITT Grupo 4*.

Ejemplo 6.-

Estudiemos la redundancia de los datos asociados a la Tabla 2 bitonal usando rachas respecto a una codificación sin compresión.

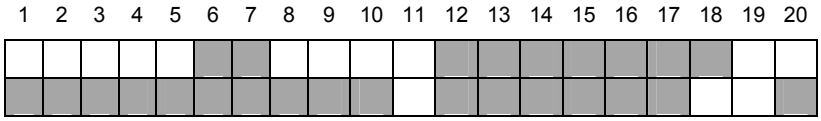


Figura 27.- Imagen bitonal que presenta rachas de pixeles.

Sin compresión, usando un dígito por cada píxel, los datos de la imagen son:

`0,0,0,0,0,1,1,0,0,0,0,1,1,1,1,1,1,1,0,0` -> tamaño = 20

`1,1,1,1,1,1,1,1,1,1,0,1,1,1,1,1,1,0,0,1` -> tamaño = 20

Mientras que la codificación mediante rachas de estas dos líneas, tomando la norma de comenzar por blanco, queda como[15]:

`5,2,4,7,2` -> tamaño = 5

`0,10,1,6,2,1` -> tamaño = 6

[14] Consultative Committee on International Telephone and Telegraphy. Comité que ha estandarizado varios algoritmos de compresión (CCITT-Grupo 3, CCITT-Grupo 4...) con vistas a su transmisión en formato digital. Ahora se conoce como UIT.

[15] Realmente en este ejemplo no se consigue compresión si se observa que sólo es necesario un bit por píxel en la codificación original, frente al entero que se necesita luego. Pero en un ejemplo de mayor tamaño si se apreciaría el efecto de la compresión.

Obteniendo unos valores para C_R y R_D:

$$C_R = \frac{(20+20)}{(5+6)} = \frac{40}{11} = 3.6363...$$

$$R_D = 1 - \frac{1}{\frac{40}{11}} = 0.725$$

Así, según esta forma de codificación, el 72'5% del tamaño de los datos representados por el código original son redundantes.

Redundancia visual

Como se ha explicado en el capítulo 1, la imagen percibida por el ojo no se corresponde exactamente con la imagen física. Por ejemplo, se ha estudiado que el humano no es capaz de distinguir entre dos niveles de gris parecidos cuando hay involucrados niveles de contraste elevados. Así, desde el punto de vista de nuestra percepción, cierta información puede considerarse menos importante y por tanto redundante.

Idealmente, los métodos de eliminación de redundancia visual modifican la imagen de manera visualmente imperceptible con el objeto de obtener una representación que permita mayor compresión. Esta modificación de la imagen supone una pérdida de información irreversible respecto de la imagen original, y por ello estos métodos se conocen como *compresores con pérdida* o con error (ver Figura 28). Esta pérdida constituye la diferencia fundamental con los métodos precedentes, ya que en aquéllos no se eliminaba información alguna, simplemente se reducía la cantidad de datos necesarios para representarla.

El ejemplo más característico de algoritmo de compresión visual es el definido por el estándar *JPEG*. Este algoritmo ha contribuido de manera fundamental a la difusión de imágenes por Internet, pues el ancho de banda de los canales de comunicación impone restricciones importantes de tamaño, que sólo se consiguen usando este tipo de compresión. Los estándares de vídeo *MPEG* y *MPEG2* (usados por los reproductores *DVD*) y el formato *JPEG2000* también son ejemplos de algoritmos de compresión visual.

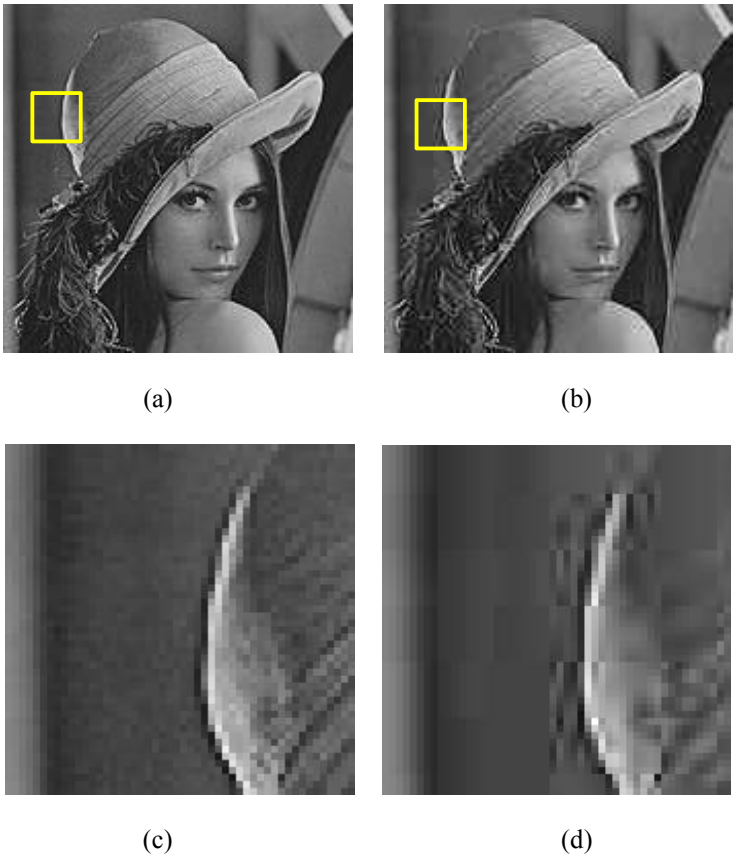


Figura 28.- Efecto de la compresión JPEG, donde se aprecian los artefactos que introduce la compresión con pérdida. (a) imagen de Lena sin comprimir ocupa 30Kb; (b) comprimiendo con el algoritmo JPEG el tamaño se reduce a 3Kb. (c) detalle del sombrero sin comprimir; (d) detalle del sombrero tras comprimir.

Básicamente, el algoritmo JPEG divide la imagen *I*, que se desea comprimir, en un conjunto de regiones cuadradas de igual tamaño *R*, aproximando los valores de los bits de cada una de estas regiones con un tipo de función conocida como *Transformada Discreta de Cosenos* (2.1). Para cada una de las regiones *R*, de dimensiones 8x8, se construye una matriz *B* (también de tamaño 8x8) aplicando la transformada. Partiendo de esta matriz *B* es posible reconstruir la región original mediante la *Transformada Discreta Inversa de Cosenos* (2.2). La transformada de cosenos tiene la propiedad de que condensa el máximo posible de información en el menor número de variables posibles. Así, unas pocas variables

de la matriz *B* contienen la mayor parte de la información necesaria para reconstruir la imagen dentro de la región *R*. La compresión se consigue porque no se almacena la matriz *B* tal cual, sino una simplificación *B'* de la misma (eliminando o cuantizando valores poco significativos). Lógicamente, cuanto más parecida sea *B'* a la matriz *B* más fiel será el parecido de la región interpolada a la original.

$$B(k_1,k_2) = C_{k_1} C_{k_2} \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} 4 \cdot I(i,j) \cdot \cos\left[\frac{p \cdot k_1}{2 \cdot N}(2i+1)\right] \cdot \cos\left[\frac{p \cdot k_2}{2 \cdot M}(2j+1)\right]$$

(2.1)

$$I(i,j) = \frac{1}{4} \sum_{k_2=0}^{N-1} \sum_{k_0=0}^{M-1} C_{k_1} C_{k_2} B(k_1,k_2) \cdot \cos\left[\frac{p \cdot k_1}{2 \cdot N}(2i+1)\right] \cdot \cos\left[\frac{p \cdot k_2}{2 \cdot M}(2j+1)\right]$$

(2.2)

Siendo $C_k = \frac{1}{\sqrt{2}}$ para k = 0, y $C_k = 1$ para el resto.

Ejemplo 7.-

Almacenando solamente los 16 coeficientes más significativos de la matriz *B* se puede interpolar una región de 8x8 bits, consiguiendo un parecido visual tal que nuestro sistema visual es incapaz de distinguirlas a simple vista. Obteniendo unos valores para C_R y R_D:

$$C_R = \frac{(8 \times 8)}{(16)} = \frac{64}{16} = 4$$

$$R_D = 1 - \frac{1}{4} = 0.75$$

Deduciéndose por tanto que el 75% del tamaño de los datos representados por el código original son redundantes. En el próximo tema, se profundizará en las propiedades de la transformada de cosenos cuando se estudie la transformada de Fourier.

2.2.3 Formatos comerciales de representación

Existen multitud de formatos de ficheros de imágenes de tipo mapa de bits. Se puede hablar de ficheros tipo *BMP*, *TIFF*, *GIF*, *JFIF*, *PGM*... Cada uno ofrece ciertas ventajas que otros formatos pueden no contemplar. La Tabla 2 recoge las principales características de algunos de estos formatos.

Formato	Color Real	Paleta	Grises	Bitonal	Compresión	Origen	Multi-Imagen
Bitmap	SI	SI	SI	SI	Run-Length	Windows	NO
TIFF	SI	SI	SI	SI	JPG, LZW, Runs, CCITT4, CCITT3, PackBits	Estándar	SI
JFIF	SI	NO	SI	NO	JPEG	Estándar	NO
PCX	NO	SI	NO	NO	Propia	Windows	NO
PGM	NO	NO	SI	NO	NO	Unix	NO
GIF	NO	SI	SI	SI	LZW	Estándar	SI

Tabla 2.- Diferentes formatos para ficheros gráficos y características principales.

Un ejemplo de formato comercial: El BITMAP de Windows

El *Bitmap* es el formato estándar dentro del sistema operativo Windows. Un Bitmap es una estructura usada para crear, manipular (rotar, mover, representar...) y almacenar imágenes.

Hay dos tipos de Bitmaps en Windows: los dependientes del dispositivo (DDB) y los independientes del dispositivo (DIB). Los DDB se crean específicamente para su presentación en un dispositivo determinado. Son más compactos pues sólo tienen la información precisa para que la imagen pueda representarse correctamente sobre ese dispositivo concreto. Por otro lado los DIB contienen información en la cabecera que los hacen relativamente independientes del dispositivo donde se van a presentar, por ello los DIB son más portables. Desgraciadamente esto conlleva un considerable desperdicio de recursos, ya que se almacena más información de la estrictamente necesaria para representar la imagen en un dispositivo concreto.

Los Bitmaps DIB, almacenados en disco, constan de una cabecera de fichero (el BITMAPFILEHEADER), una cabecera de imagen (el

BITMAPINFOHEADER), una tabla de colores (la RGBQUAD), y el conjunto de datos (referencias a la tabla de colores) que forman la imagen. La unión del BITMAPINFOHEADER y la tabla RGBQUAD forma una estructura llamada BITMAPINFO.

- Listado 1 –

```
typedef struct tagBITMAPFILEHEADER { // bmfh
        WORD    bfType;
        DWORD   bfSize;
        WORD    bfReserved1;
        WORD    bfReserved2;
        DWORD   bfOffBits;
} BITMAPFILEHEADER;

typedef struct tagBITMAPINFO { // bmi
   BITMAPINFOHEADER bmiHeader;
   RGBQUAD          bmiColors[1];
} BITMAPINFO;

typedef struct tagBITMAPINFOHEADER{ // bmih
   DWORD  biSize;
   LONG   biWidth;
   LONG   biHeight;
   WORD   biPlanes;
   WORD   biBitCount
   DWORD  biCompression;
   DWORD  biSizeImage;
   LONG   biXPelsPerMeter;
   LONG   biYPelsPerMeter;
   DWORD  biClrUsed;
   DWORD  biClrImportant;
} BITMAPINFOHEADER;

typedef struct tagRGBQUAD { // rgbq
    BYTE    rgbBlue;
    BYTE    rgbGreen;
    BYTE    rgbRed;
    BYTE    rgbReserved;
} RGBQUAD;
```

La estructura BITMAPFILEHEADER contiene datos sobre el fichero como el tamaño del fichero en bytes y la distancia del primer byte del fichero al primer byte de los datos de la imagen (*offset* de la imagen). Mientras que la BITMAPINFOHEADER contiene información sobre la imagen: el ancho y el alto en píxeles, el formato de color, si la imagen está o no comprimida, el tamaño de la imagen en bytes, la resolución original de la imagen y el número de planos necesarios para representarla (1 para bitonales, grises o paletas, 3 para imágenes RGB), el número de bits por plano necesarios para representar un color (8 bits para

color real y para grises; 1 bit para bitonales; 1, 2, 4, 8 ó 16 para paletas). Por último, la tabla de colores RGBQUAD define una paleta de colores en donde cada color se define por la intensidad para cada una de las componentes RGB.

Para la creación de Bitmaps dentro del sistema Windows se pueden utilizar las funciones de la API estándar de Windows (`CreateBitmap`, `CreateBitmapIndirect...`).

2.3. Relaciones básicas entre píxeles

A continuación, se definen ciertas relaciones que se establecen entre los píxeles de una imagen en blanco y negro con niveles de intensidad.

2.3.1 Relaciones de proximidad

Dependiendo de la situación de los píxeles y de los valores que tienen, se definen ciertas relaciones de *vecindad* y *conectividad* que se describen a continuación.

Vecindad

Para todo punto p de coordenadas (x, y) se dice que un píxel q pertenece a sus 4-*vecinos* y se escribe $q \in N_4(p)$ si y sólo si q tiene coordenadas:

$$(x\text{-}1,y) \text{ ó } (x,y\text{-}1) \text{ ó } (x\text{+}1,y) \text{ ó } (x,y\text{+}1)$$

Para todo punto p de coordenadas (x, y) se dice que un píxel q pertenece a sus 8-*vecinos* y se escribe $q \in N_8(p)$ si y sólo si q tiene coordenadas:

$$(x\text{-}1,y) \text{ ó } (x,y\text{-}1) \text{ ó } (x\text{+}1,y) \text{ ó } (x,y\text{+}1) \text{ ó } (x\text{-}1,y\text{-}1) \text{ ó } (x\text{-}1,y\text{+}1) \text{ ó } (x\text{+}1,y\text{-}1) \text{ ó } (x\text{+}1,y\text{+}1)$$

	B	A	B	
	A	p	A	
	B	A	B	

Figura 29.- Los 4-vecinos de p son los puntos A. Los 8-vecinos de p son los puntos A y B.

Conectividad

Se ha visto que una imagen se asimila a una matriz cada uno de cuyos elementos es un píxel. Entre los píxeles de esta matriz se puede definir una relación que define dos píxeles como conectados cuando son vecinos y sus valores son similares desde algún punto de vista. Formalmente, se define un conjunto V que representa los valores compatibles para que dos píxeles que sean vecinos se diga que están conectados:

V = {Niveles de gris que definen conectividad}

Se dice que dos píxeles p y q con valores en V están 4-conectados si q pertenece a $N_4(p)$.

Se dice que dos píxeles p y q con valores en V están 8-conectados si q pertenece a $N_8(p)$.

El uso de la 8 conectividad puede dar lugar a ciertas ambigüedades en ciertos análisis de conectividad. Para eliminar esta ambigüedad se define la m-conectividad. Se dice que dos píxeles p y q con valores en V están m-conectados si [$q \in N_4(p)$] ó [$q \in N_8(p)$ y $\forall x / x \in (N_4(q) \cap N_4(p))$ x no tiene valores en V]. Es decir están m-conectados si están cuatro conectados o si están 8 conectados y no tienen ninguna 4 vecino común 4 conectado.

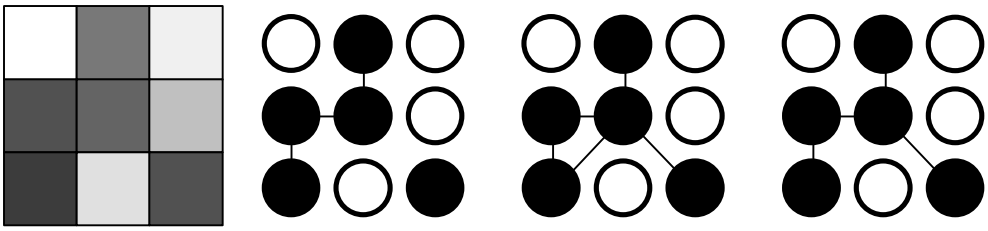
Para imágenes bitonales V puede ser el conjunto {1} o el conjunto {0}. Para imágenes en niveles de gris, con 256 niveles, V puede tener diferentes configuraciones según el interés esté en unos niveles o en otros (por ejemplo V={0,1,....,127} para obtener los elementos oscuros, o V={0,1,....,64} para obtener los muy oscuros, o V={200,201,...255} para los claros, etc.).

Ejemplo 8.-

La Figura 30 representa las relaciones de conectividad 4, 8 y m para V={0,1.....128} de la imagen A, que está definida por la siguiente matriz:

$$\text{Imagen A} = \begin{pmatrix} 255 & 120 & 240 \\ 80 & 100 & 220 \\ 60 & 225 & 80 \end{pmatrix}$$

Se aprecia que según la conectividad-4 el píxel de la esquina inferior derecha no está 4-conectado con el resto, mientras que la conectividad-8 y la conectividad-m si lo consideran conectado al resto de píxeles. Además, mientras que el píxel de la esquina inferior izquierda está *m*-conectado sólo a otro píxel está 8-conectado a dos píxeles.



(a) (b) (c) (d)

Figura 30.- Ejemplo de tipos de conectividad: (a) corresponde a una imagen en niveles de gris. (b), (c) y (d) representan en negro los píxeles de (a) que están dentro de un V determinado y muestran la relación de conexión entre ellos: (b) 4-conexión, (c) 8-conexión, (d) *m*-conexión.

Camino

Un camino desde el píxel p, de coordenadas (x, y), al píxel q, de coordenadas (s, t), es una secuencia de píxeles distintos de coordenadas:

$$(x_0, y_0), (x_1, y_1), \ldots , (x_n, y_n)$$

Donde $(x_0, y_0)=(x, y)$ y $(x_n, y_n)=(s, t)$ y (x_i, y_i) está conectado a (x_{i-1}, y_{i-1}) y siendo la n la longitud del camino. Se puede hablar de 4, 8 y *m*-caminos dependiendo del tipo de conexión involucrada.

Componente conexa

Para todo píxel p de una imagen, el conjunto de los píxeles hasta los que hay una camino desde p se dice que forman su *componente conexa*. Además se cumple que dos componentes conexas distintas tienen conjuntos de píxeles disjuntos.

2.3.2 Relaciones de distancia

La distancia más usual entre dos píxeles es la d*istancia geométrica* o d*istancia euclídea*. Así, se define la distancia entre el punto *p* de coordenadas (*x*, *y*) y el punto *q* de coordenadas (*s*, *t*) como:

$$d(p,q) = \sqrt{(x-s)^2 + (y-t)^2}$$

Nótese que siempre que sólo sea importante desde un punto de vista comparativo, esto es para comprar distancias entre puntos, se puede prescindir del cálculo de la raíz cuadrada, lo que redundará en una mayor velocidad de cálculo.

Otra distancia usual es la *distancia Manhattan* o *distancia del taxista*, que se define entre los mismos puntos *p* y *q* como:

$$d(p,q) = |x-s| + |y-t|$$

También puede citarse la distancia del *tablero de ajedrez* o *distancia chessboard* que se define como:

$$d(p,q) = \max(|x-s|, |y-t|)$$

Nótese que con la distancia Manhattan sólo los vecinos 4 conexos de un píxel están a distancia unidad, mientras que con la distancia de tablero de ajedrez todos los vecinos 8 conexos están a la distancia unidad.

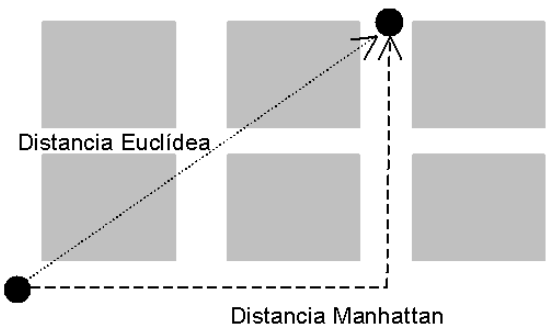


Figura 31.- Representación gráfica de la distancia Manhattan y de la distancia euclídea entre dos puntos.

2.4. Conclusiones al capítulo

Tras el estudio de este capítulo se comprende la importancia de la adecuada elección de los dispositivos de captura, su configuración y el formato de representación en un sistema informático. Se ha visto que, en general, esta elección depende del uso que se vaya a dar a la información capturada.

También se ha introducido algunos conceptos de las imágenes digitales que acompañarán al lector a lo largo del resto de capítulos.

2.5. Bibliografía del capítulo

[GW93] caps. 2 y 5,

[Esc01] cap. 3,

[Bax94] cap. 6,

[Mic96]

Capítulo 3 Filtrado y Realzado de Imagen

En este capítulo se tratan las operaciones y transformaciones que se aplican sobre las imágenes digitales en una etapa de procesamiento previa a las de segmentación y al reconocimiento. Su objeto es mejorar o destacar algún elemento de las imágenes, de manera que las etapas posteriores sean posibles o se simplifiquen.

Todas las operaciones que se van a describir en este capítulo se pueden explicar desde la perspectiva ofrecida por la *teoría de filtros*. Un *filtro* puede verse como un mecanismo de cambio o transformación de una señal de entrada a la que se le aplica una función, conocida como *función de transferencia*, para obtener una señal de salida. En este contexto se entiende por señal una función de una o varias variables independientes. Los sonidos y las imágenes son ejemplos típicos de señales.

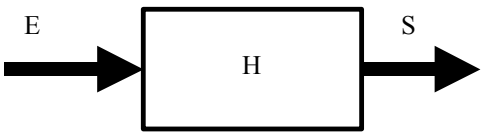


Figura 32.- Esquema de funcionamiento de un filtro.

En el diagrama anterior se representa el esquema general de funcionamiento de un filtro, siendo *E* la función de entrada, *S* la de salida y *H* la función de transferencia del filtro. Todas estas señales y funciones pueden ser discretas o continuas, y aunque en el tratamiento de imágenes se procesan señales y funciones discretas, suele recurrirse al caso continuo para explicar sus comportamientos, ya que sobre las funciones continuas es posible emplear herramientas más potentes de cálculo matemático.

En el resto del capítulo se describe las principales operaciones que se puede realizar sobre las imágenes digitales. La mayoría de las explicaciones se realizarán, por simplicidad sobre imágenes en niveles de gris. Su extensión a imágenes en color (RGB) suele consistir en repetir el tratamiento que se describe para cada una de las componentes de color. En los casos en que esto no sea posible se indicará el procedimiento adecuado.

3.1. Operaciones básicas entre píxeles

La operaciones directas sobre píxeles se pueden clasificar en operaciones aritmético-lógicas y operaciones geométricas.

3.1.1 Operaciones aritmético-lógicas

Estas operaciones son, con diferencia, las más usadas a cualquier nivel en un sistema de tratamiento de imágenes, ya que son las que se utilizan para dar valores a los píxeles de las imágenes. Las operaciones básicas son:

- *Conjunción*.- Operación lógica AND entre los bits de dos imágenes.
- Disyunción.- Operación lógica OR entre los bits de dos imágenes.
- *Negación*.- Inversión de los bits que forman una imagen.
- *Suma*.- Suma de los valores de los píxeles de dos imágenes.
- *Resta*.- Resta de los valores de los píxeles de dos imágenes.
- *Multiplicación*.- Multiplicación de los valores de los píxeles de una imagen por los de otra.

- *División*.- División de los valores de los píxeles de una imagen entre los de otra.

Se ha visto que en imágenes en niveles de gris se suele utilizar el valor 255 para representar el blanco y el 0 para el negro. Así, la operación de conjunción entre negro y blanco da como resultado negro. Si para el negro se utilizase 255 y para el blanco 0 los resultados de las operaciones conjunción y disyunción estarían intercambiados. En la Figura 33 se pueden apreciar los resultados de diferentes operaciones sobre las imágenes en niveles de gris *A* y *B*. Sobre imágenes en color los resultados serían similares.

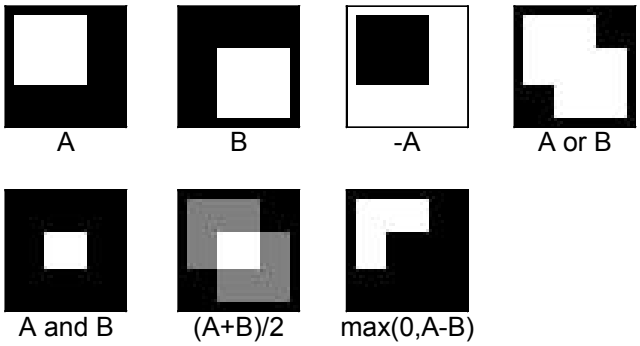


Figura 33.- Ejemplos de operaciones aritméticas y lógicas. Los píxeles a negro corresponden a bits a 0, los blancos a bits a 255.

Cuando se realizan operaciones aritméticas se debe tener la precaución de verificar que el resultado *R* de una operación cae dentro del dominio de valores permitidos. En caso contrario se puede dividir el valor *R* por un factor que consiga que el resultado pertenezca al dominio deseado. Si se desea que los valores no salgan de un rango [*m* ... *M*] se puede ajustar el resultado usando las funciones *máximo* y *mínimo* de dos valores. Para ello se toma como resultado definitivo el máximo entre *R* y el valor mínimo permitido *m* y luego el mínimo con el máximo permitido *M*, consiguiendo de esta forma que los valores nunca salgan del rango especificado.

Al implementar algoritmos que realizan operaciones aritmético-lógicas es una buena idea utilizar múltiplos del tamaño de la palabra del procesador en las transacciones con memoria, a fin de minimizar el número de operaciones de acceso a la misma. También se deben implementar los algoritmos de forma tal que en

accesos consecutivos se referencien posiciones de memoria cercanas, de manera que el criterio de proximidad referencial, usado por las memorias caché, permita un rendimiento óptimo.

3.1.2 Operaciones geométricas

Si se expresan los puntos en coordenadas homogéneas, todas las transformaciones se pueden tratar mediante multiplicación de matrices. Las operaciones geométricas más usuales son:

- *Traslación*.- Movimiento de los píxeles de una imagen según un vector de movimiento. La siguiente transformación muestra el resultado de trasladar el punto (x, y) según el vector (d_x, d_y), obteniendo el punto (x', y').

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & d_x \\ 0 & 1 & d_y \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

- *Escalado*.- Cambio del tamaño de una imagen. La siguiente transformación muestra el resultado de escalar el punto (x, y) en un factor (s_x, s_y), obteniendo el punto (x', y').

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

- *Rotación*.- Giro de los píxeles de una imagen en torno al origen de coordenadas. La siguiente transformación muestra el resultado de rotar el punto (x, y) un ángulo θ, obteniendo el punto (x', y').

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} \cos(q) & -sen(q) & 0 \\ sen(q) & \cos(q) & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

Las operaciones geométricas matriciales se pueden agrupar multiplicando las matrices. De esta forma, por ejemplo, es posible tener una única matriz que realice un desplazamiento, un giro, otro desplazamiento, y un reescalado en un solo paso. Al realizar esta composición de operaciones se debe recordar que el producto de matrices no cumple la propiedad conmutativa.

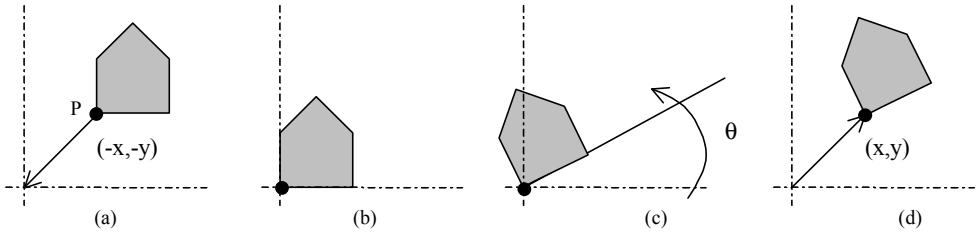


Figura 34.- Ejemplo de rotación. (a) Imagen original que se desea rotar en torno al punto *P* de coordenadas (*x*,*y*); (b) resultado de la primera traslación; (c) resultado del giro; (d) resultado final.

Ejemplo 9.-

Para rotar un ángulo q la casa de la Figura 34 respecto al punto *P*, de coordenadas (*x*, *y*), se debe crear la matriz que lleva el punto *P* al origen de coordenadas, luego la matriz que realiza el giro respecto de este origen, y luego la matriz que desplaza el punto *P* a su posición original. Esto se consigue con la siguiente matriz.

$$\mathrm{T}(x,y)\cdot \mathrm{R}(q)\cdot \mathrm{T}(-x,-y)=$$

$$=\begin{pmatrix}1 & 0 & x\\ 0 & 1 & y\\ 0 & 0 & 1\end{pmatrix}\cdot\begin{pmatrix}\cos(q) & -sen(q) & 0\\ sen(q) & \cos(q) & 0\\ 0 & 0 & 1\end{pmatrix}\cdot\begin{pmatrix}1 & 0 & -x\\ 0 & 1 & -y\\ 0 & 0 & 1\end{pmatrix}=$$

$$=\begin{pmatrix}\cos(q) & -sen(q) & x(1-\cos(q))+ysen(q)\\ sen(q) & \cos(q) & y(1-\cos(q))-xsen(q)\\ 0 & 0 & 1\end{pmatrix}$$

3.2. Operaciones sobre el histograma

Un diagrama de barras, en el que sobre el eje de abscisas se representan los diferentes valores que pueden tomar los píxeles de una imagen y en el eje de ordenadas el número de píxeles que se encuentran en una imagen para ese valor de cuantización, se conoce como *histograma* de los niveles de cuantización de la imagen, o simplemente histograma de la imagen. Se debe notar que los histogramas no dicen nada sobre la disposición espacial de los píxeles. Un histograma es una forma de representación de imágenes en la que se produce pérdida de información. A partir del histograma de una imagen es imposible deducir la imagen que lo originó. Esto ocurre porque aunque una imagen sólo puede tener un histograma, imágenes diferentes podrían tener el mismo histograma.

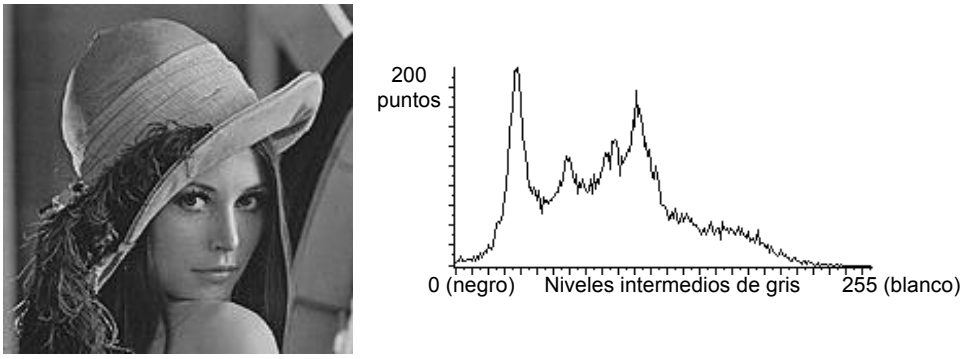


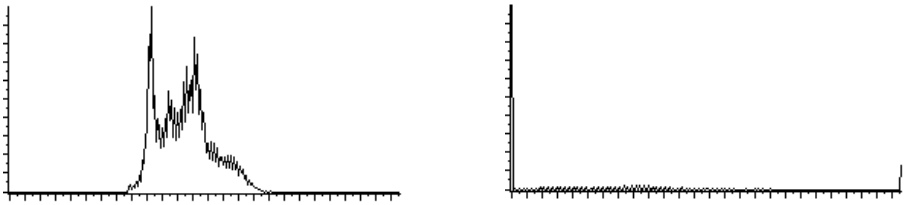
Figura 35.- Imagen en niveles de gris de Lena y su correspondiente histograma.

El histograma de una imagen en niveles de gris proporciona información sobre el número de píxeles que hay para cada nivel de intensidad (ver Figura 35). En imágenes en color RGB se usan 3 histogramas, uno por cada componente de color. En el caso de imágenes de paleta, el histograma, si bien se puede calcular, tiene una utilidad menos evidente.

Al construir histogramas, para evitar la dependencia entre el número de píxeles o el número de niveles de cuantización y el tamaño del histograma, suelen normalizarse los ejes entre 0 y 1. Esta es la razón por la que en los ejes no suelen aparecer las unidades.

Se conoce como *rango dinámico* de una imagen al conjunto de todos los posibles valores que efectivamente se encuentran presentes en una imagen.

Cuando el rango dinámico de una imagen es pequeño, generalmente, se trata de imágenes pobres en contraste, en las cuales se desaprovechan los recursos de captura. También puede darse el caso contrario, que ocurre cuando el histograma posee altos valores en los extremos de la escala, teniendo forma de "U", en este caso se dice que la imagen está saturada.



(a) (b)

Figura 36.- (a) histograma de una imagen con poco contraste. (b) histograma de una imagen saturada.

El análisis del histograma de una imagen permite conocer detalles sobre la calidad de la misma y del proceso de captura que se ha utilizado para obtenerla. Así, las imágenes de calidad suelen tener un rango dinámico amplio y no saturado. Un rango dinámico pobre implica que la imagen contiene poca información. Las imágenes muy saturadas o poco saturadas contienen menos información que las no saturadas, y por ello no son deseables para tareas de reconocimiento. También es importante para las tareas de reconocimiento que las imágenes tengan un alto nivel de contraste (sin llegar a estar saturadas), ya que esto implica que los detalles discriminantes se perciben con claridad.

Cuando la calidad de una imagen digital es pobre, bien porque esté saturada, bien porque su nivel de contraste sea bajo, la mejor opción consiste en repetir la captura y variar los parámetros del dispositivo de captura o las condiciones de iluminación de la escena. Así, aunque sobre una imagen una vez digitalizada se pueden realizar filtrados que visualmente produzcan efectos de aumento de contraste, disminución de la saturación o realzado de bordes estos cambios normalmente sólo implican una perdida de la información que contiene la imagen. Nunca un filtrado digital aumenta la información de una imagen, en el mejor de los casos no altera la información presente en la misma. Sin embargo estos filtrados pueden resultar útiles para destacar elementos de la imagen que se necesitan en las etapas de reconocimiento.

3.2.1 Aumento y reducción de contraste

Las modificaciones del histograma se pueden visualizar eficazmente mediante las *funciones de transferencia del histograma*. Estas funciones corresponden a curvas, acotadas en abcisas y ordenadas entre 0 y 1, que se definen a partir de unos valores de entrada I_E unos valores de salida I_S. En la Figura 37 se presentan tres ejemplos de funciones de transferencia: la función lineal, la función cuadrado y la función raíz cuadrada. La función de transferencia lineal (a) no introduce modificación alguna sobre el histograma, al coincidir exactamente los niveles de intensidad de la entrada y de la salida. La función cuadrado produce un oscurecimiento general de la imagen (en la figura (b) se aprecia que el rango entre 0 y 0'5 se hace corresponder con el rango entre 0 y 0'25 que es más oscuro). Por último, la función raíz cuadrada (c) produce un aclarado general de la imagen.

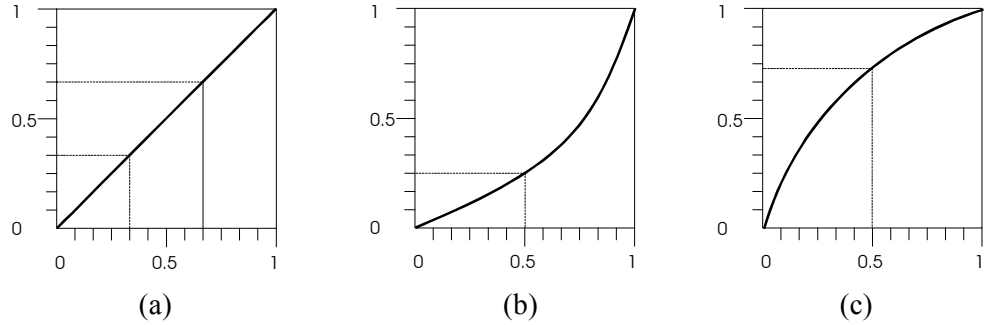


Figura 37.- De izquierda a derecha las funciones lineal, cuadrado y raíz cuadrada.

En el primer capítulo se definió contraste como la diferencia de intensidad pronunciada. Ahora, en este contexto, se puede hablar de alto contraste en una imagen digital en niveles de gris si sobre el histograma se aprecia masas separadas. En este contexto una buena medida podría ser la desviación típica del histograma.

Una función de transferencia que aclare los niveles claros y oscurezca los más oscuros, conseguirá sobre el conjunto de la imagen un efecto visual de aumento de contraste. Una función tal se puede obtener componiendo una función de transferencia del histograma que hasta el valor de 0'5 se comporte como la función cuadrado y que en adelante se comporte como la función raíz. En la Figura 38 se ha representado esta función de transferencia. La función (b) de la misma figura produce el efecto contrario, esto es una disminución del contraste.

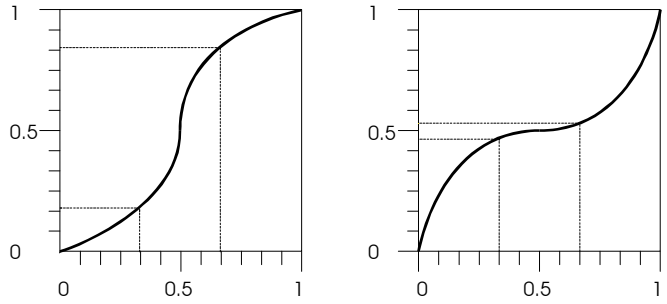


Figura 38.- Funciones de transferencia para aumento y reducción de contraste.

En general, una función de transferencia con una pendiente inferior a la unidad produce un efecto de reducción de contraste. Esto se debe a que concentra los valores de las intensidades de un rango *R* en un rango más pequeño *R'*. Por otro lado una función de transferencia con una pendiente superior a la unidad produce un efecto de aumento de contraste por razones análogas.

Las imágenes de Lena de la Figura 39 en las que se ha aumentado y reducido el contraste han sido obtenidas aplicando estas funciones de transferencia. Así, al aplicar el filtro de aumento de contraste se aprecia en su histograma que se ha saturado los tonos claros y los oscuros, mientras que se ha reducido la densidad en la parte central del histograma. Esto quiere decir que, proporcionalmente, hay muchos más píxeles asociados a los tonos blancos y negros que al resto de tonos.

El filtro de reducción de contraste produce una imagen en la que se aprecia una reducción del rango dinámico del histograma. Esto también significa que la imagen contiene menos información que otra en la que el rango dinámico sea más amplio, ya que se ha homogeneizado zonas de la imagen que antes eran diferentes.

Aunque todas las transformaciones de histograma suelen expresarse matemáticamente con fórmulas que hay que aplicar sobre cada píxel (x, y), en la práctica, suelen tabularse. Es decir, se usa una tabla para acceder al valor correspondiente a la transformación en vez de realizar el cálculo cada vez. De esta forma, se aceleran las operaciones, más teniendo en cuenta el reducido número de niveles de intensidad involucrados (normalmente 256).

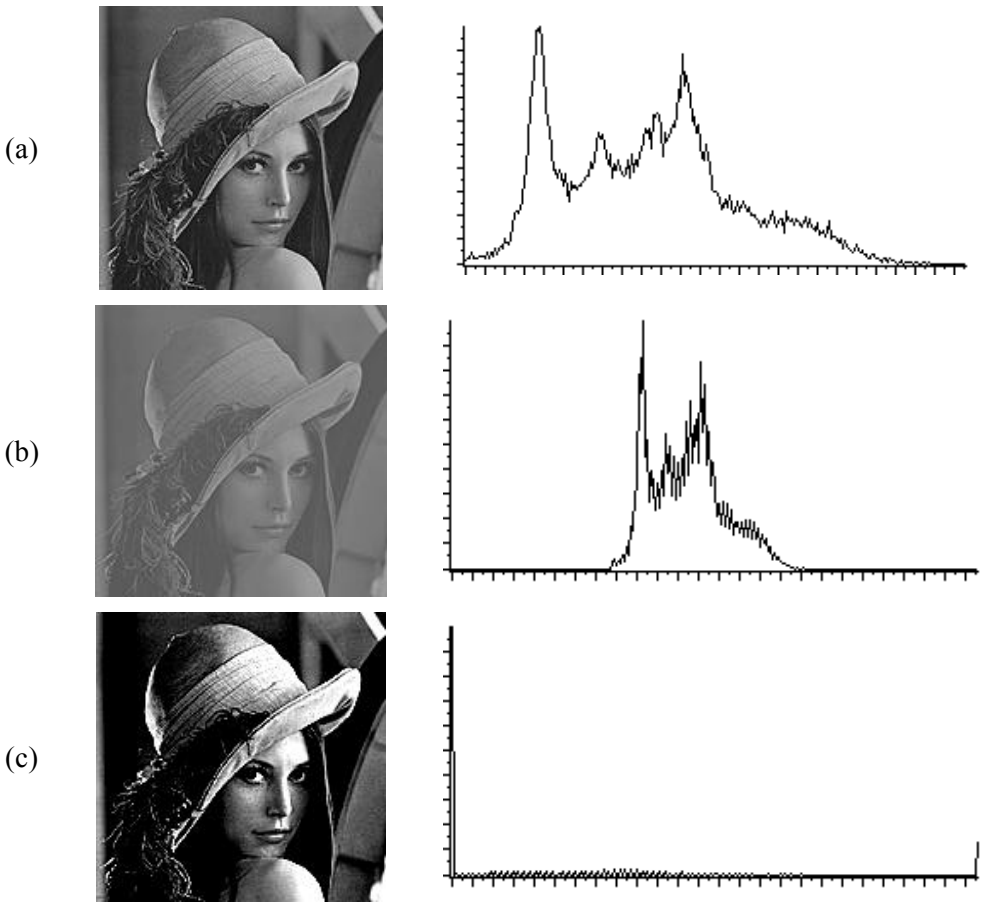


Figura 39.- Transformaciones del histograma sobre la imagen de Lena: (a) imagen original con su correspondiente histograma; (b) resultado de una operación de disminución de contraste; (c) aumento de contraste.

3.2.2 Ecualizado del histograma

El proceso de ecualizado tiene por objetivo obtener un nuevo histograma, a partir del histograma original, con una distribución uniforme de los diferentes niveles de intensidad.

Al transformar una distribución continua cualquiera en una distribución uniforme se está maximizando la cantidad de información que contiene. Y aunque ya se ha dicho que en el caso discreto es imposible aumentar la cantidad de

información, el ecualizado del histograma mejora la calidad visual de imágenes saturadas. Este efecto se debe a que se cambian los valores de intensidad de las zonas saturadas, en las que originalmente existen objetos que no se distinguen adecuadamente al inspeccionar visualmente la imagen.

Para exponer el ecualizado del histograma se modifica ligeramente la representación del histograma para asimilarla a la de una función de densidad de probabilidad. Así, los niveles de intensidad serán una variable aleatoria *R* que varía entre 0 y 1.

Esta transformación consiste en normalizar el número de intensidades a valores entre 0 y 1 (igual que previamente) y en dividir cada elemento del histograma por el número de píxeles de la imagen (para que su suma sea 1). Estas transformaciones consiguen que el área del histograma normalizado sea igual a la unidad.

Se ha dicho que el objetivo del ecualizado es transformar la distribución del histograma P_R en una distribución uniforme P_S. Como el área bajo P_R será igual al área bajo P_S, y el área bajo P_R es igual a 1, P_S debe ser una distribución uniforme con la forma P_S=1.

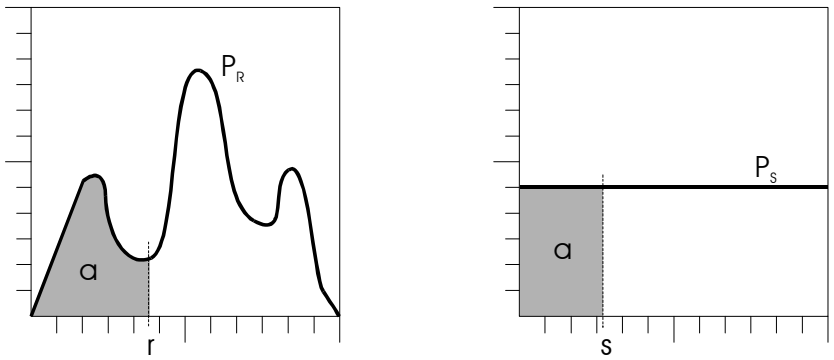


Figura 40.- Si *s* es el valor transformado de *r*, el proceso de ecualizado exige que el área a la izquierda de *r* debe ser la misma que la que hay a la izquierda de *s*.

Además, cualquier transformación del histograma cumple que el área a la izquierda de un punto *r* sobre la distribución original P_R es igual al área a la izquierda del correspondiente punto transformado *s* en la distribución P_S. Es decir, el área (a) bajo P_R de la Figura 40 debe ser igual al área (a) bajo P_S si el nivel original *r* se corresponde con *s* en el histograma ecualizado.

Todas estas restricciones se escriben como:

$$\int_0^r P_R(l)dl = \int_0^s P_S(l)dl = \int_0^s dl = s \tag{3.1}$$

Pasando al caso discreto se tiene que $P_R(r)$, la probabilidad de que un píxel tenga la intensidad r, se expresa como:

$$P_R(r) = \frac{n_r}{n} \tag{3.2}$$

Donde n es el número total de píxeles en la imagen, y n_r el número de píxeles con nivel de intensidad r. Por ello si en la fórmula (3.1) se cambia la integral por un sumatorio y se sustituye (3.2) se obtiene:

$$s(r) = \sum_{j=0}^{r} \frac{n_j}{n} \tag{3.3}$$

Para ecualizar la imagen de 3x4 de la Figura 41 (a), que posee 6 niveles de gris, se debe en primer lugar normalizar los niveles de intensidad para que tomen valores entre 0 y 1, obteniendo la imagen (b) y el histograma (c) de la Figura 41.

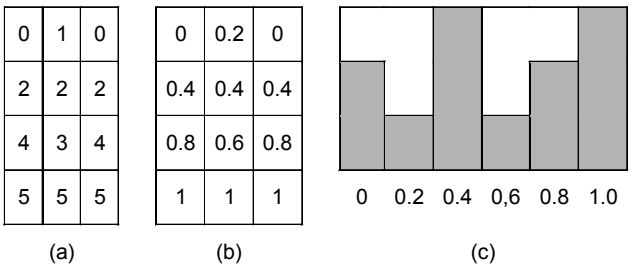


Figura 41.- (a) imagen original; (b) imagen con valores normalizados entre 0 y 1; (c) histograma de (b).

Aplicando (3.3) se obtiene:

$s(0'0) = 2/12 = 0'16$ $\quad$ $s(0'6) = 7/12 = 0'58$

$s(0'2) = 3/12 = 0'25$ $\quad$ $s(0'8) = 9/12 = 0'75$

$s(0'4) = 6/12 = 0'5$ $\quad$ $s(1'0) = 12/12 = 1$

La siguiente tabla muestra como debe realizarse la conversión de los niveles de intensidad. Estos valores han sido obtenidos al aplicar redondeo al más próximo, y en caso de igualdad hacia abajo, sobre *S*.

Nivel original	Nivel Ecualizado
0	0.16 ≈ 0.2
0.2	0.25 ≈ 0.2
0.4	0.5 ≈ 0.4
0.6	0.58 ≈ 0.6
0.8	0.75 ≈ 0.8
1.0	1.0 ≈ 1.0

Tabla 3

Cambiando estos valores se obtiene la imagen (a) y el histograma (c) de la Figura 42. Se aprecia que la transformación no ha devuelto un histograma uniforme. Esto se debe a la naturaleza discreta de los elementos, que impide dividir un nivel de intensidad concreto en varios distintos, y que también impide que aparezcan valores no enteros de intensidad.

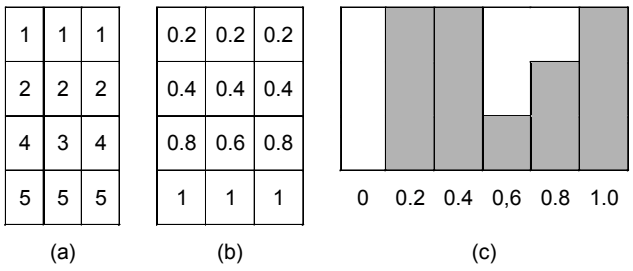


Figura 42.- (a) Resultado de la ecualización de la Figura 41; (b) histograma ecualizado.

En la Figura 43 se aprecia el efecto de una ecualizado sobre la imagen de Lena.

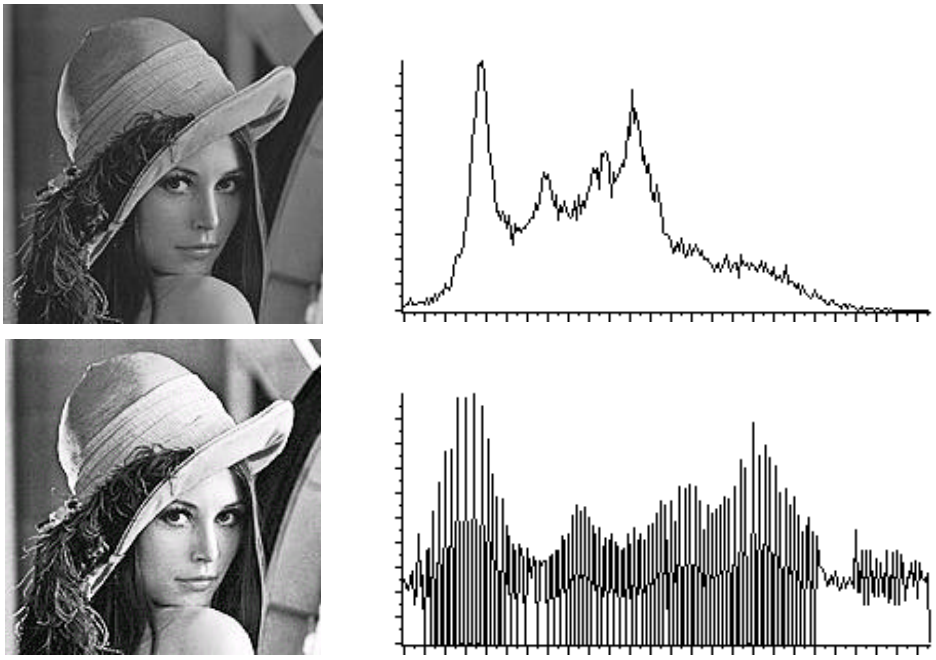


Figura 43.- Ecualizado del histograma sobre la imagen de Lena: (a) imagen original con su correspondiente histograma; (b) ecualizado del histograma.

3.3. Operaciones en el dominio de la frecuencia

Se ha visto que una imagen digital es una representación que se refiere directamente a la intensidad luminosa de puntos del espacio. Se dice que una imagen digital es una representación en el dominio del espacio. Existen otros tipos de representaciones, que contienen la misma información, pero que no están en el dominio del espacio. Es el caso de las representaciones en el dominio de la frecuencia.

Las representaciones en el dominio de la frecuencia, explican cómo se repiten ciertos patrones de una imagen, y con ello consiguen representar la información de tal imagen. Esta representación es especialmente útil, ya que teniendo la frecuencia de repetición de los elementos que componen una imagen,

se pueden apreciar y alterar directamente elementos como el ruido, los bordes, las texturas, etc.

3.3.1 Transformada de Fourier

Las *series de Fourier* son unas herramientas matemáticas especialmente útiles para describir fenómenos periódicos y para aproximar funciones no lineales.

Las series de Fourier se basan en las series trigonométricas. Se llama serie trigonométrica de coeficientes $\{a_n\}$ y $\{b_n\}$ a una serie funcional de la forma:

$$\frac{a_0}{2} + \sum_{n=1}^{\infty} a_n \cos(nx) + \sum_{m=1}^{\infty} b_m \operatorname{sen}(mx)$$

Se llama serie de Fourier de la función f(x) definida en el intervalo [$-p$, p] a la serie trigonométrica que tiene de coeficientes:

$$a_n = \frac{1}{p}\int_{-p}^{p} f(x)\cos(nx)dx \qquad \forall n = 0,1,2,...$$

$$b_m = \frac{1}{p}\int_{-p}^{p} f(x)\operatorname{sen}(mx)dx \qquad \forall m = 1,2,3...$$

En adelante se entenderá por coeficiente cada uno de los pares (a_1, b_1), (a_2, b_2)... Así el coeficiente enésimo define la amplitud de las series de cosenos y senos de frecuencia enésima.

Se puede demostrar que una serie trigonométrica definida en base a estos coeficientes converge a la función $f(x)$ que le da origen, salvo a lo sumo en un número finito de puntos. Así, una serie de Fourier puede verse como la suma de un conjunto de funciones sinusoidales de diferentes frecuencias, promediada por unos coeficientes, con el objetivo de aproximarse a una función $f(x)$. Estos coeficientes evalúan qué peso tiene cada una de las funciones sinusoidales a la hora de construir la función $f(x)$. Así, el conjunto de señales sinusoidales, debe verse como una base en el dominio de la frecuencia. Al conjunto de coeficientes correspondientes a la serie de Fourier de una función se le denomina *transformada de Fourier* de la función.

En la Figura 44 se ha calculado la transformada de Fourier de una función cuadrada (a). Se ha obtenido un conjunto de coeficientes, en los que la parte b_m es cero y a_n toma valores 1, 2/p ,0, 2/3p,0, 2/5p...

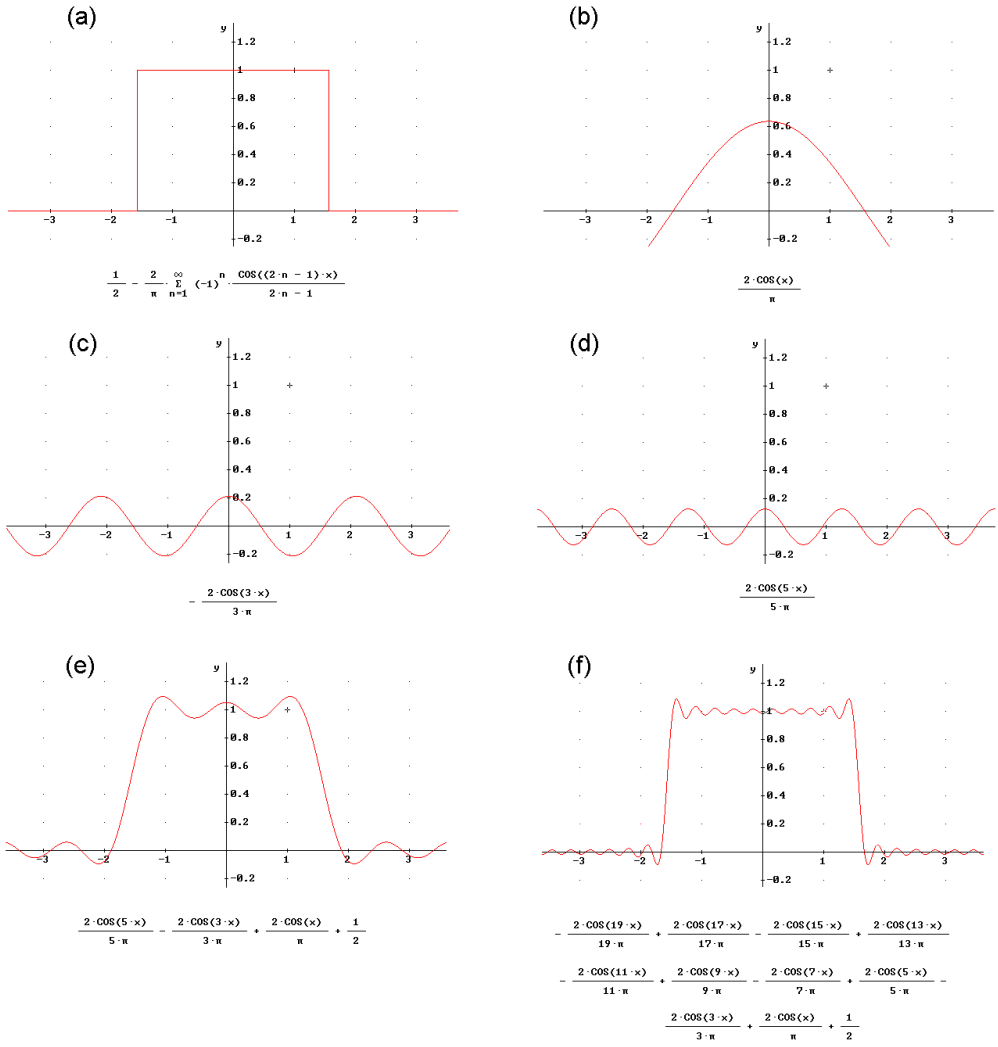


Figura 44.- En esta figura se presenta el resultado del cálculo de varios coeficientes de la transformada de Fourier de una función (a). Las gráficas (b), (c) y (d) presentan las señales sinusoidales correspondientes a los 3 primeros coeficientes de Fourier. La figura (e) corresponde a la suma de esas tres primeras señales sinusoidales. Por último, (f) corresponde a la suma de las 10 primeras componentes.

Se aprecia cómo la suma promediada de las sinusoidales (b), (c) y (d) resulta en una función (e), que aproxima la función cuadrada (a). Además se aprecia que usando los 10 primeros coeficientes se obtiene (f) que se aproxima aún más fielmente a la función original. Esto se generaliza en una propiedad importante de las series de Fourier, según la cual cuanto mayor es el número de coeficientes mejor es la aproximación de la serie a la función $f(x)$.

Representación de la transformada de Fourier

El resultado de la transformada de Fourier de una función suele representarse mediante dos diagramas. Uno indica el módulo de cada coeficiente de Fourier (a_n, b_m), el otro indica su dirección o fase. Para funciones unidimensionales estos diagramas son dos histogramas, que representan respectivamente el valor del módulo y el de la fase para cada frecuencia.

Transformada discreta de Fourier

Hasta ahora se ha tratado el caso de una función continua. Sin embargo en un ordenador las señales que se utilizan son siempre discretas. El concepto básico de señal discreta o muestreada es que está constituida por un conjunto finito de N valores.

$$F = \{f(0), f(1), \ldots f(N\text{-}1)\}$$

Estos valores se obtienen de acuerdo al proceso de muestreo o digitalización que se definió del capítulo 2. Las principales propiedades que se necesitan para el desarrollo que sigue son:

- que estas muestras se hayan tomado a intervalos fijos de tiempo o espacio.
- que se hayan tomado con la resolución adecuada para asegurar que no se pierde información.

Para poder utilizar la transformada de Fourier sobre estas señales discretas es necesario realizar algunos cambios sobre su definición. En primer lugar, se debe pasar de la definición continua a una definición discreta. Esto se consigue cambiando las integrales por sumatorios. En segundo lugar, es necesario realizar un cambio de variable sobre los coeficientes (a_n, b_n), para que la función se defina en un intervalo general [0, N-1] en vez de en el intervalo [$-p$, p]. Por último, se

puede utilizar la fórmula de Euler a fin de encontrar una notación más compacta, transformado el coeficiente (a_n, b_n) en un número complejo.

Así, para una señal temporal $f(k)$ el coeficiente enésimo ab de la serie de Fourier discreta es:

$$ab(n) = \frac{1}{N}\sum_{k=0}^{N-1} f(k)\cdot e^{-j\frac{2p}{N}kn} \qquad \forall n = 0,1,...,N-1 \qquad (3.4)$$

siendo n la frecuencia de la señal transformada. En resumen, se ha pasado de tener una señal discreta en el dominio del tiempo:

$$F = \{f(0), f(1), \ldots f(N\text{-}1)\}$$

a tener una señal discreta en el dominio de la frecuencia:

$$AB = \{ab(0), ab(1), \ldots ab(N\text{-}1)\}$$

Transformada discreta inversa de Fourier

Ya que la transformada discreta de Fourier no implica pérdida de información es posible el paso inverso. Esto es, a partir de los coeficientes de la transformada de Fourier obtener la señal en el dominio del tiempo. Para el caso discreto la transformada inversa de Fourier se define como:

$$f(k) = \sum_{n=0}^{N-1} ab(n)\cdot e^{j\frac{2p}{N}kn} \qquad \forall k = 0,1,...,N-1$$

Aplicaciones de la transformada de Fourier

Si se tuviese una señal compuesta de dos señales sinusoidales, por ejemplo correspondientes al sonido resultante de la mezcla de dos notas musicales, en el dominio de la frecuencia sería fácil aislar una de la otra. Sólo habría que realizar la transformada de Fourier, localizar las dos frecuencias dominantes, poner a cero el término de la transformada de Fourier correspondiente a la nota que se desea eliminar, y hacer la transformada inversa, obteniendo la señal correspondiente a una de las notas. Este tipo de filtrado se conoce como *filtrado paso banda*. La Figura 45 muestra diferentes ejemplos de filtrado paso banda.

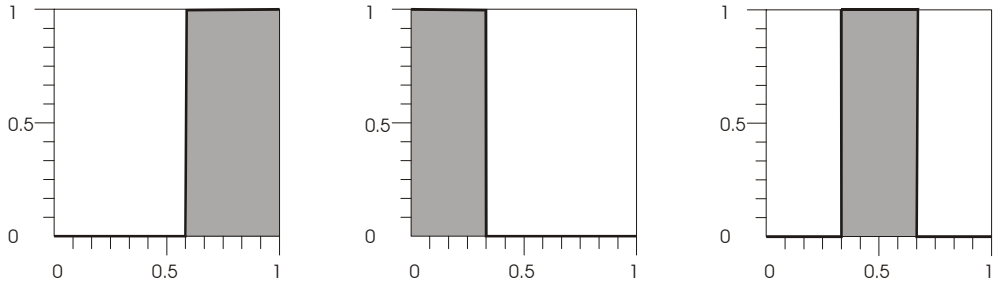


Figura 45.- De izquierda a derecha un filtro paso alto, un filtro paso bajo y un filtro paso banda.

La misma idea sirve para eliminar el ruido de alta frecuencia existente en una señal. Si se sabe que una señal que contiene cierta información puede cambiar con una velocidad máxima, los cambios que se produzcan a una velocidad mayor corresponderán a ruido. En la Figura 46 se presenta un ejemplo de una señal (a) a la que se le ha sumado un ruido de alta frecuencia (b) obteniendo (c). Calculando la transformada de Fourier, eliminando los coeficientes correspondientes a las altas frecuencias se obtiene (e), y recuperando la señal mediante la transformada inversa se obtiene (f) que es la señal original sin el ruido. Este tipo de operación que elimina el ruido de alta frecuencia se conoce como *filtrado paso bajo*.

Con un razonamiento similar es posible obtener sólo las componentes correspondientes a la alta frecuencia, en lo que se conoce como *filtrado paso alto*.

Otra aplicación se encuentra en el campo de la compresión de imágenes con pérdida. En el capítulo 2 se introdujo la transformada de cosenos para explicar la compresión con pérdida del algoritmo JPG, la cual es muy similar a la transformada de transformada de Fourier. La compresión se obtiene al guardar con mayor resolución las componentes de Fourier con valores altos, las de mayor energía[16], y guardar con poca resolución las que tengan valores bajos. Al hacer esto se consigue reducir el tamaño de los datos, ya que no se precisan tantos bits como cuando se tiene que guardar todas las componentes con la misma resolución.

[16] La energía de una señal se define como $E = \sum_{k=0}^{N-1} |f(k)|^2$

Además, para un amplio rango de imágenes, se ha comprobado que la señal obtenida al realizar la transformada inversa de estas componentes cuantizadas, se aproxima, visualmente, de manera notable a la señal original.

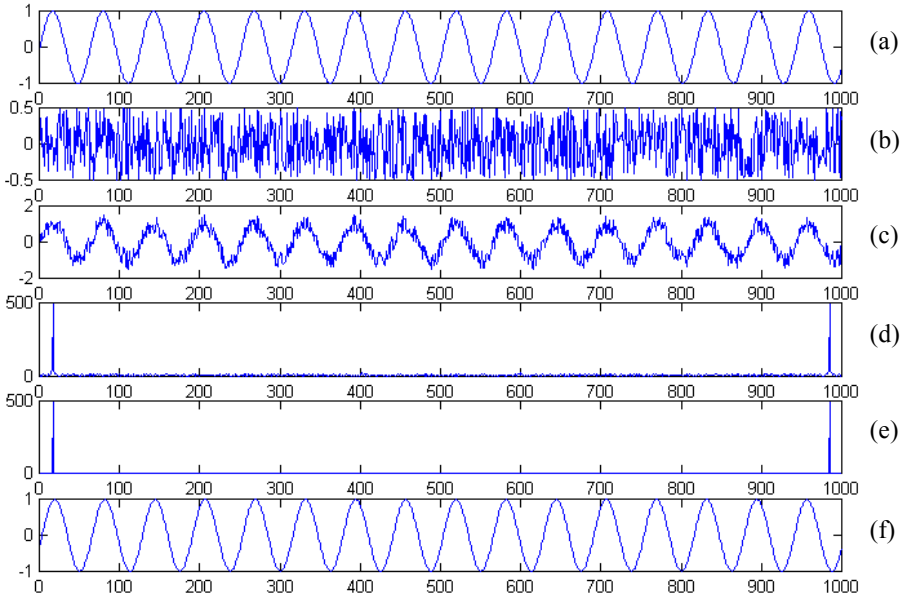


Figura 46.- (a) Señal que contiene ciertos datos, (b) ruido aleatorio, (c) la señal de datos más el ruido, (d) representación de los módulos de los coeficientes de Fourier, (e) representación de los módulos tras eliminar los que tenían poca energía, (f) el resultado de la transformada inversa de Fourier sobre los coeficientes modificados corresponde a la señal inicial sin ruido.

Transformada rápida de Fourier

El cálculo de los coeficientes de Fourier es muy costoso. La ecuación (3.4) revela que hay que hacer un bucle para calcular cada coeficiente. Por tanto el cálculo de todos los coeficientes exige un doble bucle y por eso su complejidad es $O(n^2)$. Además las operaciones de división y multiplicación involucradas en el cálculo son muy costosas. Es por esto que la optimización del cálculo de la transformada de Fourier es importante si se desea hacer un uso intensivo de ella.

La *transformada rápida de Fourier* (*FFT*[17]) no es otra cosa que una manera rápida de calcular la transformada discreta de Fourier, gracias a la aplicación de algunas técnicas matemáticas y a algunas optimizaciones en su implementación. En los siguientes párrafos se analizan las optimizaciones que utiliza la FFT para calcular de manera más eficiente los coeficientes de Fourier.

La primera optimización consiste en trasladar el cálculo de la exponencial fuera del sumatorio. Esto se puede hacer precalculando los N posibles productos. Así, partiendo de la fórmula original se ve que el término $-j2\pi/N$ es constante, mientras que $k \cdot n$ cambia en cada iteración. Se aprecia además que cada vez que $k \cdot n$ sobrepasa el valor de N-1, el resultado de la exponencial es el mismo (ya que esta exponencial es una función periódica de periodo N). Precalculando la parte exponencial y almacenándola en un array llamado PRE, la formula (3.4) queda como:

$$ab(n) = \frac{1}{N}\sum_{k=0}^{N-1} f(k) \cdot PRE(nk \bmod \mathrm{N}) \qquad \forall n = 0,1,...,N-1$$

La segunda optimización que realiza la FFT se basa en la observación de que se pueden computar por separado los términos pares de los impares de la serie de Fourier. Así, los términos pares tienen la forma:

$$ab(2n) = \frac{1}{N}\sum_{k=0}^{\frac{N}{2}-1} SUM(k) \cdot PRE(nk \bmod \mathrm{N}) \qquad \forall n = 0,1,...,N/2-1 \quad (3.5)$$

donde:

$$SUM(k) = f(k) + f(k + \frac{N}{2})$$

Mientras los términos impares tienen la forma:

[17] *Fast Fourier Transform* en inglés.

$$ab(2n+1) = \frac{1}{N}\sum_{k=0}^{\frac{N}{2}-1} DIF(k)\cdot PRE(nk \bmod \mathrm{N}) \qquad \forall n = 0,1,...,N/2-1 \quad (3.6)$$

donde:

$$DIF(k) = f(k) - f(k+\frac{N}{2})e^{j\frac{2p}{N}k}$$

Esta optimización hace que para el cálculo de un término de la serie sean necesarios sólo $N/2$ productos, frente a los N que se necesitaban antes, reduciendo con esto el tiempo de cómputo a la mitad.

Si el número de valores N de la función a aproximar es potencia de dos, la idea de calcular de manera independiente términos pares e impares se puede aplicar de manera sucesiva (este tipo de algoritmos se conocen con el nombre de "divide y vencerás"), sobre las fórmulas (3.5) y (3.6). Esta optimización reduce la complejidad del algoritmo a $O(n\cdot\log(n))$.

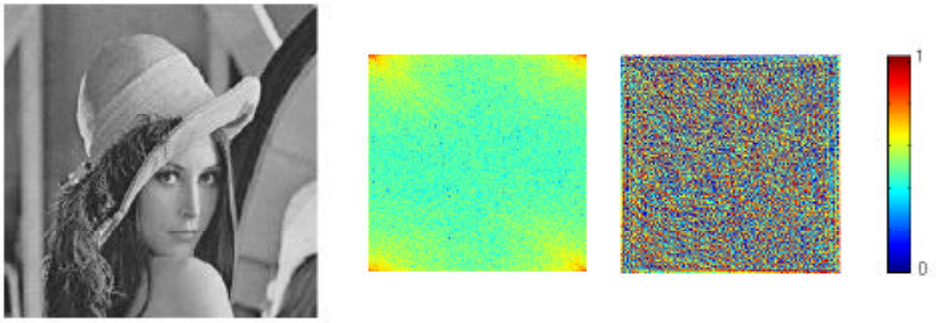
Transformada de Fourier de imágenes digitales

Para el tratamiento de imágenes digitales en niveles de gris $I(x,y)$ se debe ampliar la definición de la transformada de Fourier para funciones bidimensionales. Así, los coeficientes $I_C(n, m)$ se calculan según:

$$I_C(n,m) = F(I(x,y)) = \frac{1}{N}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} I(x,y)\cdot e^{-j\frac{2pxn}{N}}\cdot e^{-j\frac{2pmy}{N}}$$
$$\forall n,m = 0,1,...N-1$$

Debe apreciarse que esta definición sólo resulta aplicable sobre imágenes cuadradas.

En la Figura 47 se presenta las matrices correspondientes a los módulos y a las fases de los coeficientes de Fourier de la transformada de la imagen de Lena. Los valores de estas matrices han sido normalizados entre 0 y 255, y han sido representados en falso color.



(a) (b) (c)

Figura 47.- En esta figura se muestra una imagen de Lena (a) sobre la que se realiza una transformada de Fourier, obteniéndose (b) y (c), que corresponden respectivamente a la representación matricial de los módulos y de las fases de los coeficientes de Fourier normalizados entre 0 y 1 y en falso color.

La matriz de módulos contiene la información relativa a los valores de intensidad de la imagen (las amplitudes de las sinusoides). La matriz de fase contiene la información relativa a la posición de los píxeles (la posición de los flancos de subida y bajada de las sinusoides). Es por eso que si se realiza la transformada inversa teniendo en cuenta sólo la matriz de fase se obtiene una imagen parecida al trazado de los bordes de la imagen. Por otro lado, la realización de la transformada inversa teniendo en cuenta sólo la matriz de módulos proporciona una imagen de manchas con tonos parecidos a los de la imagen original.

La inversa de la transformada de Fourier de una imagen viene dada por:

$$I(x,y) = F^{-1}(I_C(n,m)) = \frac{1}{N}\sum_{n=0}^{N-1}\sum_{m=0}^{N-1} I_C(n,m)\cdot e^{-j\frac{2pxn}{N}}\cdot e^{-j\frac{2pmy}{N}}$$
$$\forall n,m = 0,1,...N-1$$

De nuevo estas transformaciones se pueden realizar sobre imágenes en color (RGB) sin más que repetir el tratamiento que se describe para cada una de las componentes de color.

La Transformada discreta de Fourier bidimensional tiene una complejidad $O(n^4)$, ya que hay que calcular (3.7) para cada píxel de la imagen.

Afortunadamente, el algoritmo FFT descrito para el caso unidimensional es aplicable al caso bidimensional. Con él, la complejidad del cálculo de los coeficientes de Fourier es de $O(n^2 \cdot \log(n))$.

3.3.2 Filtrado frecuencial

Una vez conocida la formulación de la transformada de Fourier para imágenes en niveles de gris ya es posible transformar una imagen del dominio del espacio al dominio de la frecuencia. Como ya se ha visto, una vez en el dominio de la frecuencia, es sencillo realizar filtrados que eliminen elementos que aparezcan con cierto periodo. La Figura 48 presenta la situación de los coeficientes correspondientes a las altas y a las bajas frecuencias sobre la matriz de coeficientes. Como ocurría en el caso unidimensional, las componentes cercanas a cero (en este caso las esquinas de la matriz) corresponden a las bajas frecuencias, mientras que las del centro corresponden a las altas frecuencias. En los siguientes apartados se tratan en detalle los procesos de filtrado paso bajo, filtrado paso alto y filtrado paso banda para imágenes digitales.

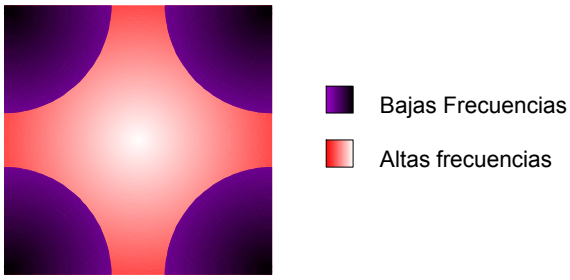


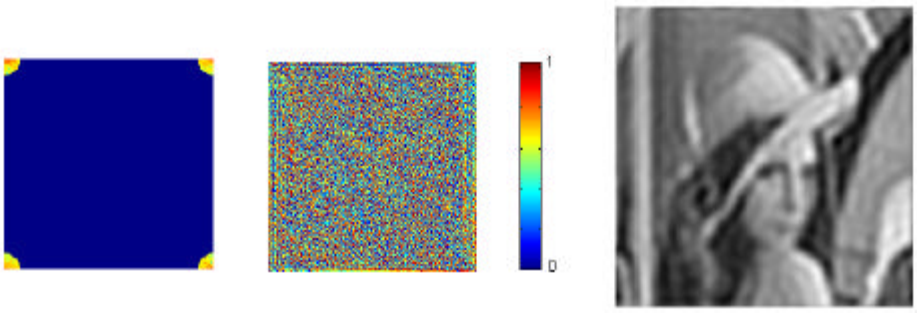
Figura 48.- Situación de los valores correspondientes a las altas y a las bajas frecuencias sobre la matriz de coeficientes de la transformada discreta de Fourier bidimensional.

Filtros paso bajo

Los filtros paso bajo son filtros que eliminan las frecuencias altas, dejando "pasar" las bajas frecuencias. Para realizar un filtrado de este tipo basta con poner a cero los módulos de los coeficientes de Fourier relativos a las altas frecuencias, dejando sin modificar los relativos a las bajas frecuencias. Es importante elegir un valor de corte adecuado a partir del cual se considera que una frecuencia es alta o baja.

En la Figura 49 se presenta la imagen de Lena de la Figura 47 sobre la que se ha aplicado un filtrado paso bajo. Sobre la matriz de módulos (a) se ha marcado

en azul los valores que han sido puestos a cero. Se observa que sólo han pasado los módulos de las frecuencias inferiores a 30 píxeles.



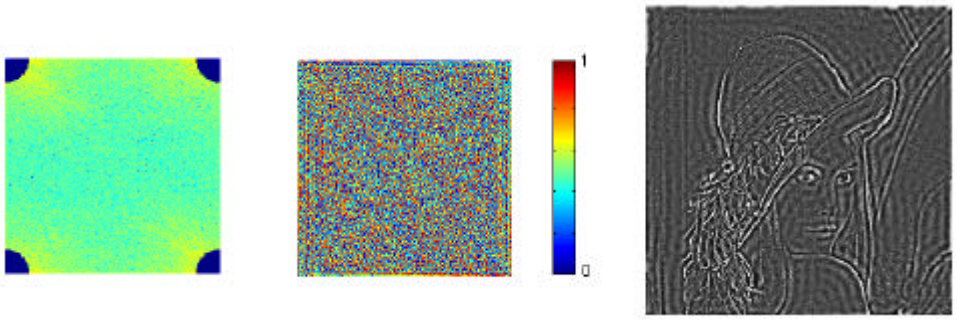
(a) (b) (c)

Figura 49.- Modificación sobre las matrices de coeficientes poniendo a cero el módulo correspondiente a las altas frecuencias (a), y manteniendo las fases (b). El resultado de la transformada inversa sobre las matrices de coeficientes modificadas corresponde a la imagen (c) donde se aprecia que los bordes han sido suavizados.

Filtros paso alto

Los filtros paso alto son filtros que eliminan las bajas frecuencias, permaneciendo las altas frecuencias. Para realizar un filtrado paso alto hay que poner a cero los módulos de los coeficientes de Fourier relativos a las bajas frecuencias, dejando sin modificar los relativos a las altas frecuencias. De nuevo hay que elegir un valor de corte adecuado a partir del cual se considera que una frecuencia es alta o baja.

En la Figura 50 se presenta la imagen de Lena de la Figura 47 sobre la que se ha aplicado un filtrado paso alto. Sobre la matriz de módulos (a) se aprecia que se ha pasado los módulos de las frecuencias superiores a 30 píxeles, el resto, en azul, está a cero. Sobre (c) se aprecia que las altas frecuencias se corresponden con los bordes de la figura de Lena, ya que los bordes corresponden a los cambios rápidos de intensidad de los píxeles.



(a) (b) (c)

Figura 50.- Modificación realizada sobre las matrices de coeficientes poniendo a cero el módulo correspondiente a las bajas frecuencias (a), y manteniendo las fases (b). El resultado de la transformada inversa sobre las matrices de coeficientes modificadas corresponde a la imagen (c) donde se aprecia sólo la información de los bordes y el ruido de alta frecuencia.

Filtros paso banda

Los filtros paso banda son filtros en los que permanece inalterado un rango (o banda) de frecuencias determinado y son eliminados los coeficientes correspondientes al resto de frecuencias. Así, los filtros paso alto y paso bajo constituyen dos casos límites del filtro paso banda.

En la imagen de la Figura 51 se ha preparado una imagen de Lena a la que se le ha añadido un ruido con estructura. Sobre la representación de la matriz de módulos se aprecia unos valores altos para las frecuencias cercanas a los puntos (0,80), (0,180), (255,80) y (255,180) que en la Figura 47 no existían. Posteriormente se ha realizado un filtrado paso banda en el que se ha eliminado dos bandas de radio 25 en torno a los puntos que concentraban las frecuencias de ruido. Sobre la figura se puede apreciar que los píxeles de las líneas que interferían han sido rellenados con valores próximos a los de sus vecinos.

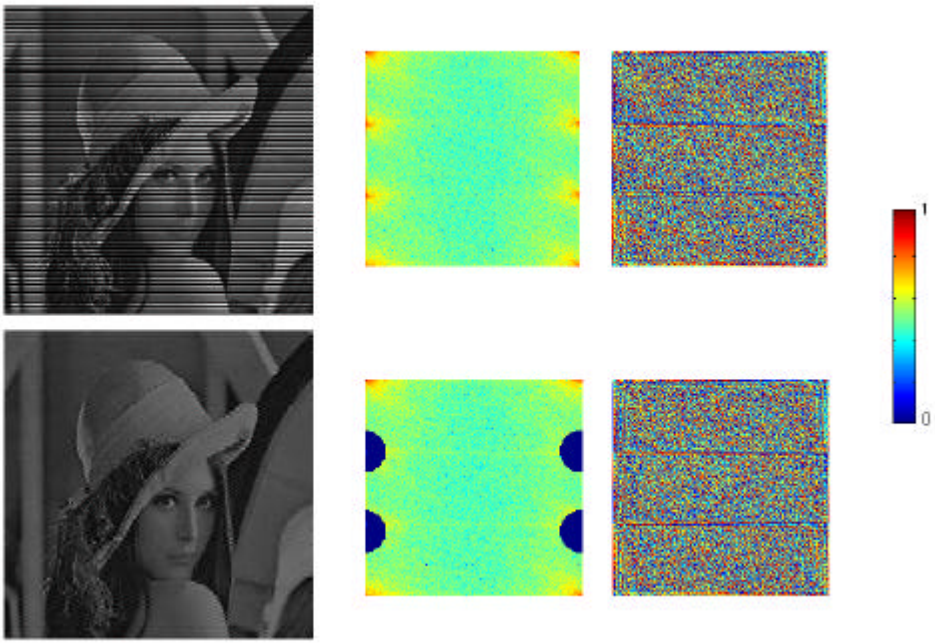


Figura 51.- Arriba la imagen de Lena a la que se le añade un ruido con estructura que consiste en la desaparición de 1 de cada 3 líneas de la imagen. Abajo el resultado de un filtrado paso banda ajustado a la zona del histograma de Fourier en el que aparecen las frecuencias que corresponden al ruido.

3.3.3 Otros operadores en el dominio de la frecuencia

Existen multitud de operadores que también operan en el dominio de la frecuencia y que no se estudian en este capítulo por motivos de espacio. En general su operativa es similar a la estudiada para la Transformada de Fourier. Sin embargo, cada uno aporta sus propias ventajas e inconvenientes. Por ejemplo, se puede citar la transformada de cosenos que se introdujo brevemente en el capítulo 2.

No se debe terminar esta sección sin citar las transformadas de *onditas* (*wavelets*) que suponen una generalización al concepto de transformada de Fourier en el que la base del espacio vectorial no son las funciones senos y cosenos sino cualquier tipo de función dentro de una amplia familia (Haar, Gausiana, Mexican Hat, etc...). Estos operadores están obteniendo gran aplicación debido a la particularización que supone elegir adecuadamente la función de la base para cada

problema concreto y a la reducción del coste computacional de los algoritmos asociados a su cálculo respecto a la transformada de Fourier. Así, por ejemplo, el nuevo estándar de compresión de imágenes *JPEG2000* usa compresión por onditas.

3.4. Filtrado espacial

Se puede demostrar matemáticamente que la transformación al dominio de la frecuencia (F), la aplicación del filtro (H) sobre los coeficientes, y la aplicación de la inversa de la transformada (F^{-1}), se puede realizar directamente en el dominio del espacio. Esta operación en el dominio del espacio se conoce como *convolución*, y exige el conocimiento de la transformada inversa (h) del filtro.

En el caso bidimensional discreto, la transformada inversa (h) del filtro resulta una matriz de tamaño NxN, que es la función de transferencia que se conoce como *función impulsional*. Así, la operación de convolución queda:

$$I'(x,y) = I(x,y) * h(x,y) = \sum_{i=-\frac{N}{2}}^{\frac{N}{2}} \sum_{j=-\frac{N}{2}}^{\frac{N}{2}} I(i,j) \cdot h(x-i, y-j)$$
$$\forall x, y = 0,1,..., N-1$$

La carga computacional de esta operación es elevada. Sin embargo es posible realizar aproximaciones a esta operación usando una función de transferencia h que tenga un número de elementos muy inferior a NxN. Estas funciones impulsionales reducidas se denominan *funciones de filtrado espacial*. El siguiente ejemplo muestra la aplicación de un filtrado espacial con una máscara de convolución de 3x3.

Ejemplo 10.-

Supóngase una función impulsional h de tamaño 3x3 igual a la matriz que sigue:

$$h = \begin{pmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{pmatrix}$$

La imagen de salida *I'* resultante de aplicar el filtrado espacial viene dada por la expresión:

$$I'(x,y) = I(x,y) * h = h_1 I(x\text{-}1, y\text{-}1) + h_2 I(x, y\text{-}1) + h_3 I(x\text{+}1, y\text{-}1) + h_4 I(x\text{-}1, y) + h_5 I(x, y) + h_6 I(x\text{+}1, y) + h_7 I(x\text{-}1, y\text{+}1) + h_8 I(x, y\text{+}1) + h_9 I(x\text{+}1, y\text{+}1)$$

Se aprecia que, para una máscara de convolución de 3x3, el valor del píxel $I'(x,y)$ tras el filtrado depende únicamente del valor del píxel $I(x,y)$ y de sus ocho vecinos antes del filtrado.

Para mantener el resultado de la operación dentro de un rango representable se puede añadir a la expresión anterior un factor de división y un factor de suma.

3.4.1 Filtros paso bajo

El filtrado paso bajo espacial se basa en el promediado de los píxeles adyacentes al píxel que se evalúa. Quizás el filtro paso bajo más simple que se puede diseñar se corresponde con una matriz de 3x3 con todos los elementos a 1. El resultado se deberá dividir por 9 para obtener valores dentro del rango de la paleta. En la figura adjunta se puede apreciar el resultado de la aplicación de este filtro.

Las siguientes matrices de convolución definen otros filtros paso bajo:

$$h = \frac{1}{10}\begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{pmatrix} \qquad h = \frac{1}{16}\begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix}$$

Otro filtro paso bajo es el *filtro de la mediana*. Éste se basa en sustituir el valor de un píxel por el de la mediana del conjunto formado por el mismo y sus ocho vecinos.

El *filtro del bicho raro* es otro ejemplo de filtro paso bajo. Consiste en comparar la intensidad de un píxel con la de sus 8 vecinos. Si la diferencia es superior a cierto umbral *U* (que debe elegirse previamente), se sustituye tal píxel

por el valor promedio de los píxeles vecinos, en otro caso se mantiene su valor de intensidad.

Tanto el filtro de la mediana, como el filtro del "bicho raro" son filtros no lineales[18], es decir, no se pueden deducir de una convolución, y por tanto no tienen equivalente en el dominio de la frecuencia.

$$\frac{1}{9}\begin{pmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{pmatrix}$$

Original Paso bajo

Figura 52.- Aplicación de un filtrado espacial paso bajo.

3.4.2 Filtros paso alto

El cálculo de la derivada direccional de una función nos habla de cómo se producen los cambios en tal dirección. Tales cambios, que se corresponden con las altas frecuencias, suelen corresponder a los bordes de los objetos presentes en las imágenes.

Partiendo de que el operador gradiente se define como:

$$\nabla(I(x,y)) = \frac{\partial I}{\partial x}\vec{u}_x + \frac{\partial I}{\partial y}\vec{u}_y$$

[18] Un operador O sobre imágenes bidimensionales se dice que es lineal si se cumple que:

$$O[k_1 \times I_1(x,y) + k_2 \times I_2(x,y)] = k_1 \times O[I_1(x,y)] + k_2 \times O[I_2(x,y)]$$

siendo k_1 y k_2 dos constantes e I_1 e I_2 dos imágenes bidimensionales.

Se definen los filtrados de convolución G_x, y G_y:

$$G_x = \frac{\partial I}{\partial x} = I(x,y) * h_1(x,y)$$

$$G_y = \frac{\partial I}{\partial y} = I(x,y) * h_2(x,y)$$

Obteniendo h_1 y h_2 mediante una aproximación a la derivada con la resta. Es decir, si se consideran los píxeles de la siguiente figura:

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

las derivadas serían:

$$\frac{\partial f}{\partial x} = z_5 - z_6 \qquad \text{y} \qquad \frac{\partial f}{\partial y} = z_5 - z_8$$

Obteniendo unas matrices de convolución h_1 y h_2.

$$h_1 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & 0 \end{pmatrix} \qquad h_2 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 0 \end{pmatrix}$$

En el caso de que no se desee considerar la dirección del vector gradiente, sino sólo su módulo, se escribe:

$$\left| I_G(x,y) \right| = \sqrt{G_x^2(x,y) + G_y^2(x,y)}$$

Con el fin de reducir la carga computacional la expresión anterior puede sustituirse por esta otra:

$$\left|I_G(x,y)\right| = \frac{1}{2}\left|G_x(x,y) + G_y(x,y)\right|$$

Una aproximación mejor al gradiente está dada por las expresiones:

$$\frac{\partial f}{\partial x} = (z_3 + 2z_6 - z_9) - (z_1 + 2z_4 - z_7)$$

$$\frac{\partial f}{\partial y} = (z_1 + 2z_2 - z_3) - (z_7 + 2z_8 - z_9)$$

Dando lugar a las matrices h_1 y h_2.

$$h_1 = \frac{1}{4}\begin{pmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{pmatrix} \qquad h_2 = \frac{1}{4}\begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix}$$

Estas matrices se conocen como *ventanas de Sobel*, que fue quien las propuso. Mediante ellas se calcula el gradiente en las direcciones horizontal y vertical. En la Figura 53 se ve cómo el resultado de aplicar h_1 sobre la imagen de Lena produce una imagen en la que aparecen los contornos horizontales de la figura de la imagen original.

Otros sencillos filtros espaciales que se pueden encontrar en la bibliografía son los de Roberts y los de Prewitt.

$$\text{Robert}: \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \text{ y } \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \qquad \text{Prewitt}: \begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix} \text{ y } \begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix}$$

$$\frac{1}{4}\begin{pmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{pmatrix}$$

Original Paso alto

Figura 53.- Filtrado paso alto de Sobel en la dirección *x* en valor absoluto. El resultado se presenta con el valor de los píxeles invertidos.

3.4.3 Filtro de la laplaciana

El operador laplaciano de una función bidimensional $I(x,y)$ es el escalar:

$$\Delta(I(x,y)) = \nabla(\nabla(I(x,y))) = \frac{\partial^2 I}{\partial x^2}\vec{u}_x + \frac{\partial^2 I}{\partial y^2}\vec{u}_y$$

Este operador, que se basa en la segunda derivada, se hace cero cuando la primera derivada se hace máximo, es decir cuando aparece un cambio de signo en la primera derivada. Su cálculo parte del de la primera derivada.

$$\begin{pmatrix} z_1 & z_2 & z_3 \\ z_4 & z_5 & z_6 \\ z_7 & z_8 & z_9 \end{pmatrix} \text{ y derivando se obtiene } \begin{pmatrix} z_1' & z_2' & z_3' \\ z_4' & z_5' & z_6' \\ z_7' & z_8' & z_9' \end{pmatrix}$$

siendo:

$$z_4' = z_4 - z_5 \qquad \text{y} \qquad z_5' = z_5 - z_6$$

y por tanto:

$$z_4'' = z_4' - z_5' = (z_4 - z_5) - (z_5 - z_6) = z_4 - 2z_5 + z_6$$

Procediendo de igual forma en la dirección vertical se obtiene la matriz de convolución de la figura adjunta.

Original

$$\frac{1}{8}\begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

Laplaciana

Figura 54.- Laplaciana de la imagen de Lena. Para su presentación se añade un factor de suma de 128.

3.5. Operaciones morfológicas

Clásicamente la morfología es una parte de la biología que estudia la forma de los animales y de las plantas. De la misma forma, la *morfología matemática* es una herramienta que ayuda a tratar problemas que involucran formas en una imagen. La morfología matemática tiene su origen en la teoría de conjuntos. Para ella las imágenes binarias son conjuntos de puntos 2D, que representan los puntos activos de una imagen, y las imágenes en niveles de gris son conjuntos de puntos 3D, donde la tercera componente corresponde al nivel de intensidad. En este apartado sólo se tratará detalladamente la morfología sobre imágenes bitonales, presentándose únicamente los operadores básicos para imágenes en niveles de gris.

3.5.1 Definiciones básicas

Sean A y B conjuntos (con las operaciones habituales entre conjuntos) de Z^2 (con las aplicaciones habituales entre vectores). Sean a de coordenadas (a_1, a_2) los puntos de A y b de coordenadas (b_1, b_2) los puntos de B. Se definen las siguientes operaciones:

Traslación de A por $X = (x_1, x_2)$, como:

$$(A)_x = \{ c / c = a + x, \forall a \in A\}$$

Reflexión de A como:

$$\hat{A} = \{ x / x = -a, \forall a \in A\}$$

Complementario de A como:

$$A^c = \{ x / x \notin A\}$$

Diferencia de dos conjuntos A y B como:

$$A - B = \{ x / x \in A \quad \text{y} \quad x \notin B\}$$

Una propiedad interesante que se deriva de las operaciones anteriores es:

$$A - B = A \cap B_c$$

Dilatación

Siendo A y B dos conjuntos en Z^2, la *dilatación* de A con B, denotada como $A \oplus B$, se define:

$$A \oplus B = \{x / x = a + b \qquad \forall a \in A \quad \text{y} \quad \forall b \in B\}$$

Es interesante notar que la dilatación cumple la propiedad conmutativa.

$$A \oplus B = B \oplus A$$

La implementación directa de la dilatación según la definición dada es demasiado costosa. La siguiente formulación, que puede demostrarse que es equivalente, da una idea de una implementación mucho más eficiente.

$$A \oplus B = \{x / (\hat{B})_x \cap A \neq f\}$$

Escrito de otra forma:

$$A \oplus B = \{x / [(\hat{B})_x \cap A] \subseteq A\}$$

El elemento *B* es el elemento que dilata a *A*, y se conoce como *elemento estructurante* de la dilatación.

Intuitivamente esta operación produce el efecto de dilatar el aspecto del elemento A usando para ello a B (ver Figura 55). La posición del elemento B respecto del eje de ordenadas es importante, ya que influye en el proceso de dilatación. Por ello suele indicarse el centro de las figuras con un punto.

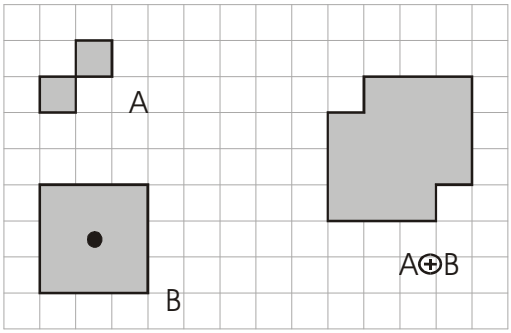


Figura 55.- Ejemplo de dilatación en el que se ha señalado con un punto negro el origen del elemento *B*.

Erosión

Siendo *A* y *B* dos conjuntos en Z^2, la *erosión* de *A* con *B*, denotada como $A\Theta B$, se define:

$$A\Theta B = \{x / x + b \in A \quad \forall b \in B\}$$

Nuevamente puede definirse con otra forma cuyo coste computacional es mucho más reducido.

$$A\Theta B = \{x / (B)_x \subseteq A\}$$

La erosión adelgaza la imagen sobre la que se aplica siendo, en un sentido no estricto, opuesta a la dilatación. Si sobre la Figura 55 se erosiona $A \oplus B$ con *B* se obtiene de nuevo *A*, aunque esto no tiene por qué ocurrir en otro caso distinto. En la Figura 56 se ha erosionado *A* con *B*.

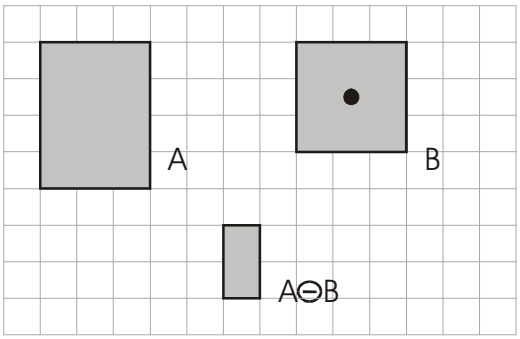


Figura 56.- Ejemplo de erosión.

Se cumple que la dilatación y la erosión son duales respecto al complemento y a la reflexión. Es decir:

$$(A \Theta B)^c = A^c \oplus \hat{B}$$

Esta propiedad se puede demostrar fácilmente con el siguiente desarrollo:

$$(A \Theta B)^c = \{x/(B)_x \subseteq A\}^c = \{x/(B)_x \cap A^c = f\}^c =$$
$$= \{x/(B)_x \cap A^c \neq f\} = A^c \oplus \hat{B}$$

Apertura

La *apertura* de *A* con *B* se define como:

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

Sus propiedades son:

- $A \circ B$ es un subconjunto de A.
- $(A \circ B) \circ B = A \circ B$.
- Si C es subconjunto de $D \Rightarrow C \circ B$ es subconjunto de $D \circ B$.

Intuitivamente la apertura de *A* con un elemento estructurante *B* equivale a determinar los puntos en los que puede situarse *B* cuando se desplaza por el interior de *A* (ver Figura 57).

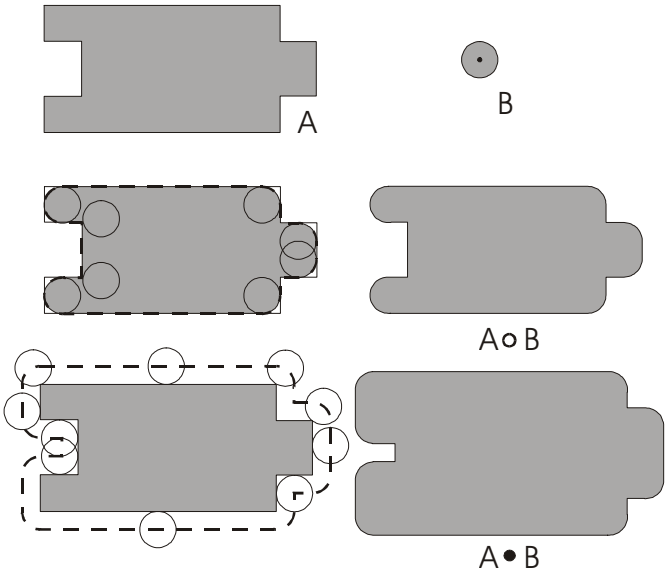


Figura 57.- Arriba se presenta la figura *A* y el elemento estructurante *B*. En medio se presenta la ejecución de la operación de apertura y su resultado. Abajo se presenta la operación de cierre.

Cierre

El *cierre* de *A* con *B* se define como:

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

Sus propiedades son:

- *A* es un subconjunto de $A \bullet B$.
- $(A \bullet B) \bullet B = A \bullet B$.
- Si *C* es subconjunto de $D \Rightarrow C \bullet B$ es subconjunto de $D \bullet B$.

Intuitivamente el cierre de *A* con un elemento estructurante *B* equivale a los puntos en los que puede estar el origen de *B* cuando se desplaza tocando al menos un punto de *A* (ver Figura 57).

Coincidencia estructural

Si el elemento *B* se define teniendo en cuenta los puntos a blanco que lo rodean, se tiene la descripción de un objeto B_1 y su entorno B_2. La operación de *coincidencia estructural*, o *Hit or Miss*, busca la parte de la imagen *A* que cumpla que los puntos activos de *B* estén en *A* y los puntos del entorno de *B* estén en A^c.

$$A * B = (A \ominus B_1) \cap (A^c \ominus B_2)$$

El elemento estructurante *B* suele definirse sobre una cuadrícula donde: un cuadro sombreado indica que el píxel pertenece a B_1, un cuadro blanco indica que el píxel pertenece a B_2, un cuadro con una cruz indica que el cuadro no debe tenerse en cuenta.

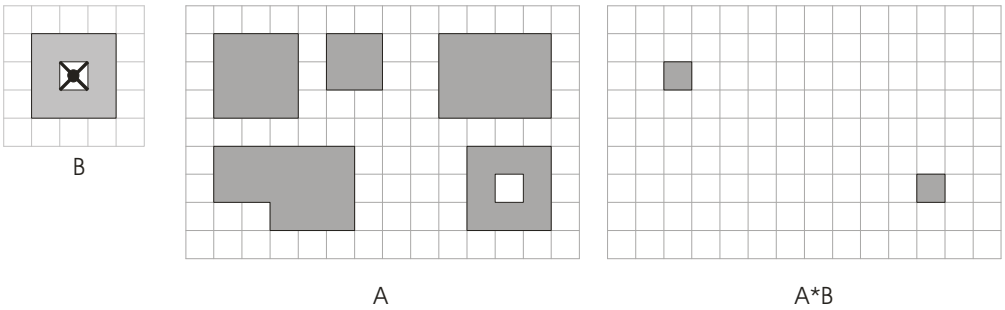


Figura 58.- Ejemplo de operación de coincidencia estructural.

3.5.2 Filtros morfológicos

Los siguientes puntos ilustran cómo emplear morfología para construir filtros.

Eliminación de ruido

Este filtro elimina los objetos de una imagen que tienen un tamaño menor que un elemento estructurante *B* determinado.

$$\text{Limpiar}(A) = (A \circ B) \bullet B$$

Este filtro elimina los objetos menores que *B* en el proceso de apertura, luego intenta recuperar la misma forma que antes con el proceso de cierre (aunque el proceso de apertura y cierre no son estrictamente inversos).

Extracción de bordes

Este filtro obtiene los bordes de una figura *A* restándole su interior.

$$\text{Bordes}\,(A) = A - (A \ominus B)$$

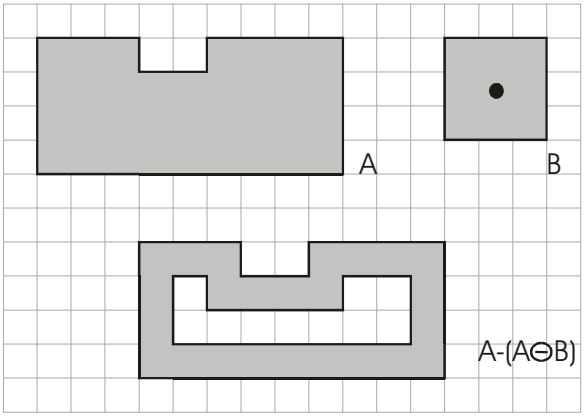


Figura 59.- Ejemplo de extracción de bordes morfológica.

Relleno de agujeros

Este filtro precisa de un proceso iterativo que concluye cuando no se producen más cambios sobre la imagen. Se parte de X_0 igual a un punto del agujero que se desea rellenar. Luego se aplica de manera iterativa:

$$X_k = (X_{k-1} \oplus B\,) \cap A^c$$

Adelgazamiento

Esta operación adelgaza los elementos de una imagen hasta que se reducen a un esqueleto interior a la misma. Para poder realizar esta operación es preciso definir la siguiente operación:

$$A \otimes B = A - (A * B)$$

Este paso, utilizando la operación de coincidencia estructural, elimina un punto de A cuando tanto B como su entorno encajan en A. Eligiendo cuidadosamente un conjunto de elementos B, que sólo deben erosionar los bordes de A, y repitiendo esta operación sucesivamente se obtiene el resultado deseado.

$$A \nabla \{B\} = (\ldots((((A \otimes B1) \otimes B2) \otimes B3) \otimes B4) \ldots \otimes Bn)$$

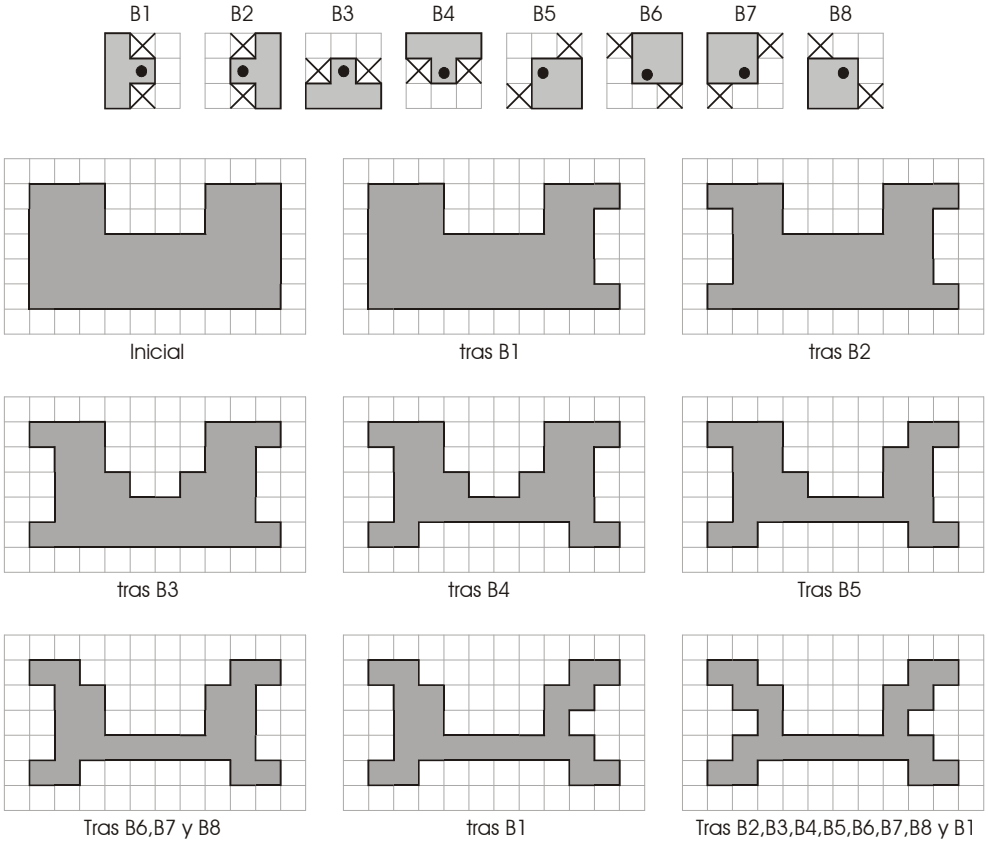


Figura 60.- Ejemplo de aplicación del algoritmo de adelgazamiento. Las máscaras definidas van erosionando por el borde la figura original, en iteraciones sucesivas.

3.5.3 Operaciones morfológicas básicas en imágenes de niveles de gris

Considerando las imágenes en niveles de gris como superficies $I(x, y)$ y tomando como elemento estructural una función $b(x, y)$ se definen las operaciones de erosión y dilatación como:

$$I(x, y) \oplus b = \max\{I(x - i, y - j) + b(i, j) \ / (x - i, y - j) \in \mathrm{Dom}(I) \text{ y } (i, j) \in \mathrm{Dom}(b)\}$$

$$I(x, y) \ominus b = \min\{I(x - i, y - j) - b(i, j) \ / (x - i, y - j) \in \mathrm{Dom}(I) \text{ y } (i, j) \in \mathrm{Dom}(b)\}$$

Las operaciones de apertura y cierre se definen igual que para el caso bitonal. Intuitivamente estas operaciones se corresponden con el desplazamiento del elemento estructurante b sobre la superficie $I(x, y)$ en la apertura y bajo la misma en el cierre.

3.5.4 Aplicaciones de la morfología matemática

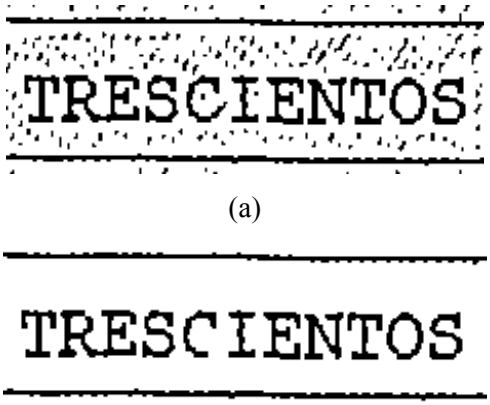
En general, la morfología matemática se puede emplear para realizar cualquier tipo de operación sobre una imagen. Se puede realizar filtrados lineales y no lineales, operaciones geométricas, aritmético-lógicas, etc. Todos los filtrados que han sido definidos en apartados anteriores encuentran una formulación morfológica. Por ejemplo, el suavizado se consigue con una apertura y un cierre, y un filtro de detección de bordes con una dilatación y la resta de una erosión.

La Figura 61 presenta un ejemplo en el que se ha usado una dilatación en blanco y negro para reconstruir un código de barras de baja calidad. Antes del filtrado era imposible la lectura del código de barras con un dispositivo convencional, tras este filtrado la lectura del código de barras es posible.

(a) (b)

Figura 61.- Ejemplo de reconstrucción de un código de barras degradado (a). Se utiliza una operación dilatación con elemento estructurante vertical de 9x1 y se obtiene (b).

La Figura 62 muestra un ejemplo de eliminación de ruido en blanco y negro mediante una erosión y una dilatación. Se aprecia que antes del filtrado (Figura 62a) era más difícil localizar los objetos que corresponden a los caracteres a reconocer que después del filtrado (Figura 62b).



(a)

(b)

Figura 62.- Ejemplo de eliminación de ruido de la figura (a) mediante una erosión y una dilatación posterior con un elemento estructural de 3x3, resultando la figura (b).

Normalmente, la aplicación directa de un conjunto de operaciones morfológicas usando concatenación de operadores genéricos es impracticable debido a su coste computacional. Por ello, suele aplicarse de manera restringida, utilizando software muy optimizado, o utilizando hardware específico. Queda no obstante la utilidad de usar la morfología para realizar primeras aproximaciones a problemas y para realizar demostraciones, gracias a su carácter intuitivo y a la formalidad matemática que aporta.

3.6. Conclusiones al capítulo

En este capítulo se ha estudiado diferentes operaciones que usadas adecuadamente facilitan las etapas posteriores de análisis de las imágenes.

Sin embargo, debe recordarse que realizando estas u otras manipulaciones nunca se gana información, sólo se promociona o se descarta información ya existente en la imagen. Por ello siempre que la calidad de una imagen sea pobre,

resultando insuficiente para el uso al que se destina, debe pensarse en la posibilidad de variar las condiciones de captura.

3.7. Bibliografía del capítulo

[GW93] caps. 3 y 4,

[JKS95] caps. 4 y 5,

[Esc01] caps. 4 y 5,

[Par97] caps. 1 y 2

Capítulo 4 Segmentación

La segmentación es un proceso que consiste en dividir una imagen digital en regiones homogéneas con respecto a una o más características (como por ejemplo el brillo o el color) con el fin de facilitar su posterior análisis y reconocimiento automático. Localizar la cara de una persona dentro de la imagen de una fotografía o encontrar los límites de una palabra dentro de una imagen de un texto, constituyen ejemplos de problemas de segmentación.

En este capítulo se estudiarán diferentes enfoques para realizar el proceso de segmentación, aunque en la práctica se demuestra que la segmentación no tiene reglas estrictas a seguir, y dependiendo del problema en cuestión, puede ser necesario idear técnicas a medida.

También se tratará, al final del capítulo, el problema de la descripción de los objetos resultantes de la segmentación. Entonces, se estudiarán métodos que permiten una descripción de los objetos segmentados independiente de la posición y de la escala de los mismos.

4.1. Conceptos básicos sobre segmentación

La segmentación debe verse como un proceso que a partir de una imagen, produce otra en la que cada píxel tiene asociada una etiqueta distintiva del objeto al que pertenece. Así, una vez segmentada una imagen, se podría formar una lista de

objetos consistentes en las agrupaciones de los píxeles que tengan la misma etiqueta.

La segmentación termina cuando los objetos extraídos de la imagen se corresponden unívocamente con las distintas regiones disjuntas a localizar en la misma. En este caso se habla *de segmentación completa* de la escena o imagen y en el caso contrario, de *segmentación parcial*. En una escena compleja, el resultado de la segmentación podría ser un conjunto de regiones homogéneas superpuestas y en este caso, la imagen parcialmente segmentada deberá ser sometida después a un tratamiento posterior con el fin de conseguir una segmentación completa.

El proceso de segmentación de una imagen depende del problema que se desee resolver. Por ejemplo, sobre una imagen de una página de texto se pueden segmentar las líneas de texto (si el objetivo es localizar la estructura de los párrafos), o las palabras y los caracteres que las forman (si se desea hacer *OCR*[19] de los mismos), o los logotipos y membretes (si se desea clasificar el documento), etc. Por ello, dentro de una misma imagen pueden buscarse diferentes segmentaciones.

En general, el proceso de la segmentación suele resultar complejo debido, por un lado, a que no se tiene una información adecuada de los objetos a extraer y, por otro, a que en la escena a segmentar aparece normalmente ruido. Es por esto que el uso de conocimiento sobre el tipo de imagen a segmentar o alguna otra información de alto nivel puede resultar muy útil para conseguir la segmentación de la imagen.

Algunos ejemplos típicos de procesos de segmentación son los de tratar de separar los caracteres que forman una palabra dentro de una imagen de un texto, los de detectar ciertos tipos de células en imágenes médicas, los de extraer los vehículos que aparecen en una imagen de una carretera, o en general cuando se trata de separar ciertos objetos de un fondo en una imagen cualquiera.

Los diferentes objetos que aparecen en una imagen pueden localizarse atendiendo a aspectos como: sus bordes o su textura. Cada una de las técnicas que se van a estudiar en este capítulo atienden a alguna de estas características, y para

[19] Optical Character Recognition, reconocimiento óptico de caracteres.

su estudio han sido englobadas en tres grupos: técnicas basadas en umbralización, basadas en detección de los bordes de los objetos y técnicas basadas en propiedades locales de las regiones.

4.1.1 La textura

Intuitivamente la textura de un objeto dentro de una imagen es el conjunto de formas que se aprecia sobre su superficie y que le dotan de cierto grado de regularidad. Una definición típica de *textura* es la siguiente: "uno o más patrones locales que se repiten de manera periódica".

Para el estudio y comparación de algoritmos sobre imágenes que presentan texturas suelen utilizarse como referencia las imágenes de Brodatz, conocidas como *álbum de Brodatz* (P. Brodatz, "Textures: A Photographic Album for Artists and Designers", Dover Publications, New York, 1966). Este álbum contiene 154 imágenes. La Figura 63 muestra algunas imágenes de este álbum.

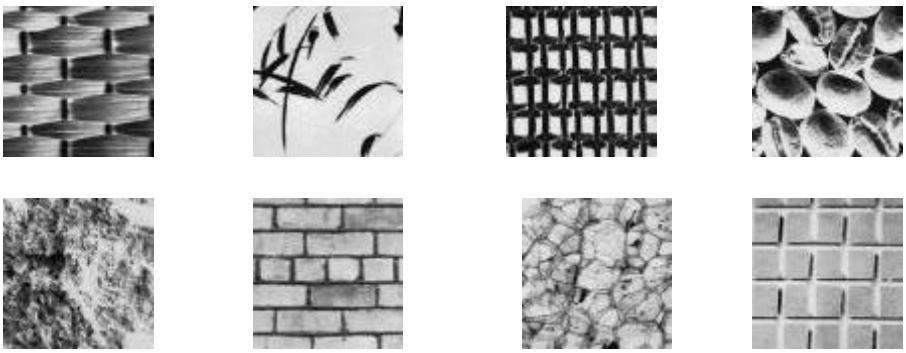


Figura 63.- Diversas imágenes del álbum de Brodatz, utilizadas en el análisis de texturas.

Existen dos enfoques para definir una textura: uno descendente ("top-down") y otro ascendente ("bottom-up"). El enfoque descendente se basa en la existencia de un elemento básico de textura, llamado *téxel*, y en una regla de formación. Esta regla define cómo y dónde se sitúan estos elementos básicos. Este enfoque funciona bien cuando la textura es bastante regular, por ejemplo en la imagen de una pared de ladrillos. El enfoque ascendente se basa en que la textura es una propiedad que se puede derivar de estadísticos (como la media y la varianza) de pequeños grupos de píxeles. Este enfoque funciona bien para texturas donde resulta difícil ver los componentes individuales, por ejemplo la textura de la

hierba o el cuarzo. No obstante, la línea divisoria entre los dos enfoques no es clara.

4.1.2 El borde

Los bordes de un objeto en una imagen digital corresponden a la línea de píxeles que separa ese objeto del fondo de la imagen. Normalmente estos bordes se corresponden con los puntos donde se producen discontinuidades en los valores de los píxeles adyacentes (cambios en el matiz o el brillo) o en conjuntos de píxeles (cambios de textura).

4.2. Segmentación basada en umbralizado

La *umbralización* es un proceso que permite convertir una imagen de niveles de gris o en color en una imagen binaria, de tal forma que los objetos de interés se etiqueten con un valor distinto de los píxeles del fondo o *background*. En adelante sólo se hablará de imágenes en niveles de gris, aunque la extensión a color es inmediata si sólo se usa una de las componentes RGB o alguna mezcla de las tres.

La umbralización es una técnica de segmentación rápida, que tiene un coste computacional bajo y que incluso puede ser realizada en tiempo real durante la captura de la imagen usando un computador personal de propósito general.

4.2.1 Umbralización fija

El caso más sencillo, conocido como *umbralización fija*, se puede usar en aquellas imágenes en las que existe suficiente contraste entre los diferentes objetos que se desea separar. Así, se puede establecer un valor fijo que marque el umbral de separación sobre el histograma. Para obtener dicho umbral se debe disponer de información sobre los niveles de intensidad de los objetos a segmentar y del fondo de la imagen. De esta forma la imagen binaria resultante $B(i, j)$ se define a partir de la imagen digital original $I(i, j)$ en función de un valor U que corresponde al umbral de separación seleccionado.

$$B(i,j) = \begin{cases} 1, \text{si } I(i,j) \geq U \\ 0, \text{si } I(\text{i},\text{j}) < U \end{cases}$$

Como puede comprenderse, la elección de un valor de umbral correcto resulta decisiva para llevar a cabo la segmentación de una imagen de manera satisfactoria. La obtención del umbral se basa en el histograma de la imagen (ver capítulo 3). Cuando en el histograma se aprecian uno o más lóbulos, éstos suelen corresponder con una o varias zonas de la imagen, las cuáles comparten niveles de intensidad similares. Estos objetos pueden ser directamente los objetos a segmentar o corresponder a partes homogéneas de objetos más complejos. Lógicamente, la transición de un lóbulo a otro se corresponde con un mínimo del histograma, siendo estos mínimos puntos sobre los que se suele umbralizar. La búsqueda de dichos mínimos (basada en el cálculo de derivadas) se encuentra dificultada por la naturaleza ruidosa del histograma. Para atenuar este problema suele aplicarse un filtro de suavizado sobre el histograma de la imagen.

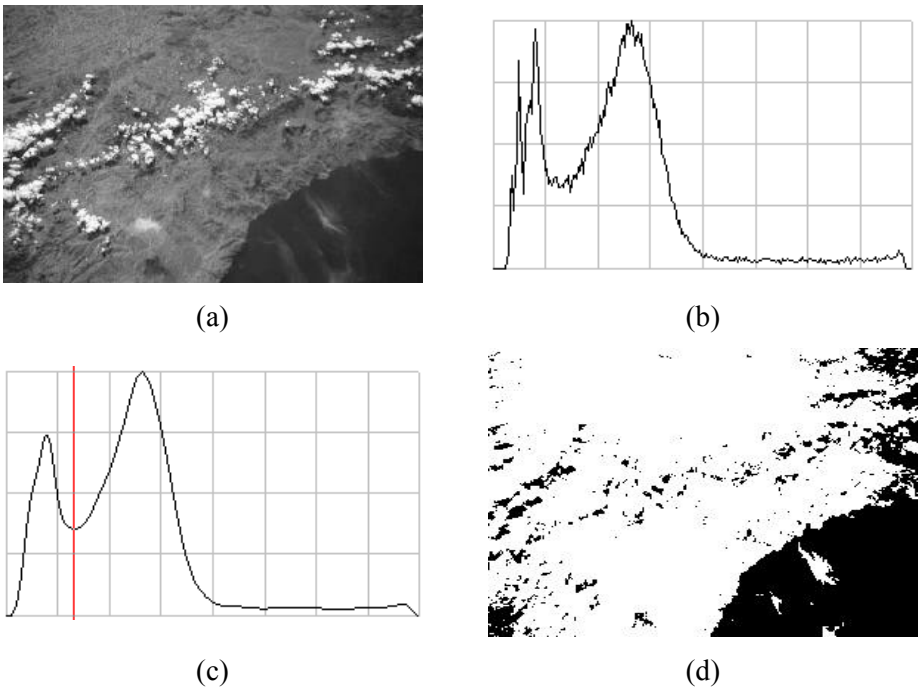


Figura 64.- La elección adecuada del umbral de binarización permite separar la tierra del mar en esta imagen de satélite. Se aprecia que la sombra de las nubes sobre la tierra y las nubes sobre el mar producen errores. Por ello es preciso algún posproceso.

La Figura 64 ilustra el proceso descrito. En ella se distinguen una zona de tierra y otra de mar en una foto de satélite (a). El histograma (b) presenta dos lóbulos que tras ser suavizados (c) usando un filtro espacial de paso bajo, muestra

un mínimo de separación en el valor 42. Eligiendo dicho mínimo como umbral de separación entre tierra y mar, se obtiene el resultado de la figura (d).

El algoritmo siguiente resume los pasos descritos para realizar una umbralización sobre el histograma $h(p)$, donde $p=(x,y)$ representa a cada píxel de la imagen que para una imagen $I(x,y)$ de niveles de gris toma valores entre 0 y 255.

- Algoritmo de localización de mínimos locales -

Paso 1.- Filtrado paso bajo de la imagen $I(x, y)$ usando una ventana de tamaño V.

$$\text{Repetir } \forall k \in \{V/2 \le k \le 255 - V/2\}: \qquad h_F(p_k) = \frac{1}{V} \sum h(p_{(k-V/2)})$$

Paso 2.- Cálculo de la primera y de la segunda derivada de h_F. Para ello se puede usar la aproximación de la derivada como resta de valores de posiciones consecutivas:

$$h_F'(k) = h_F(k+1) - h_F(k)$$
$$h_F''(k) = h_F'(k+1) - h_F'(k)$$

Paso 3.- Si $h_F'(k) \pounds U$, siendo U un umbral positivo, y $h_F''(k) > 0$, entonces hay que marcar k como candidato a mínimo local.

Paso 4.- Agrupación de candidatos a mínimos locales y umbralización basada en éstos.

4.2.2 Umbralización generalizada.

En general, la obtención de un único valor de umbral fijo no es aplicable en imágenes más complejas y además, debido a problemas de iluminación no uniforme de la escena, el estudio del histograma de la imagen no permite detectar un valor del umbral adecuado para separar los objetos del fondo o *background*. Esto lleva a considerar otros tipos de umbralización para segmentar una imagen, que resultan de generalizar la idea de umbral explicada. A continuación, se definen la *umbralización de banda*, la *multiumbralización*, la *semiumbralización* y la *umbralización adaptativa*.

Umbralización de banda

La *umbralización de banda* permite segmentar una imagen en la que los objetos (regiones de píxeles) contienen niveles de gris dentro de un rango de valores y el fondo tiene píxeles con valores en otro rango disjunto. Así,

$$B(i,j) = \begin{cases} 1 & \text{si } I(i,j) \in R \\ 0 & \text{en otro caso} \end{cases}$$

donde R representa un rango de valores correspondientes a niveles de gris que definen a los elementos a extraer de la imagen digital.

Multiumbralización

La *multiumbralización*, como su nombre indica, consiste en la elección de múltiples valores de umbral dentro del proceso, permitiendo separar a diferentes objetos dentro de una escena cuyos niveles de gris difieran. El resultado no será ahora una imagen binaria sino que los diferentes objetos (regiones) tendrán etiquetas diferentes:

$$\begin{aligned} I_S(i,j) &= 1, && \text{si } I(i,j) \in R_1 \\ &= 2, && \text{si } I(i,j) \in R_2 \\ &= 3, && \text{si } I(i,j) \in R_3 \\ &\cdots\cdots\cdots\cdots\cdots\cdots && \\ &= n, && \text{si } I(i,j) \in R_n \\ &= 0, && \text{en otro caso} \end{aligned}$$

donde $I(i, j)$ es la imagen original, $I_S(i, j)$ es la imagen segmentada y R_1, R_2, ..., R_n representan los n diferentes rangos de niveles de gris usados para umbralizar.

Semiumbralización

La *semiumbralización* persigue obtener una imagen resultado en niveles de gris, y para ello pone a cero el fondo de la imagen conservando los niveles de gris de los objetos a segmentar que aparecen en la imagen inicial. Así:

$$I_S(i,j) = \begin{cases} I(i,j) & \text{si } I(i,j) \geq U \\ 0 & \text{en otro caso} \end{cases}$$

Umbralización adaptativa

En las técnicas anteriores, el o los rangos de umbralización se consideran fijos con independencia de las características locales de la imagen considerada. En muchas imágenes donde la iluminación no es uniforme puede ocurrir que píxeles del

mismo objeto a segmentar tengan niveles de gris diferenciados. La *umbralización adaptativa* o *variable* permite resolver el problema de la segmentación haciendo que el valor del umbral varíe como una función de las características locales de la imagen.

- Algoritmo de umbralización adaptativa -

Paso 1.- Dividir la imagen original $I(i,j)$ en subimágenes $I_k(i,j)$ donde se supone que los cambios de iluminación no son tan fuertes

Paso 2.- Determinar independientemente un umbral U_k para cada subimagen $I_k(i,j)$;

Paso 3.- Si en alguna subimagen no se puede determinar su umbral, calcularlo mediante la interpolación de los valores de los umbrales de subimágenes vecinas; y

Paso 4.- Procesar cada subimagen con respecto a su umbral local.

Como ya se ha expuesto en el capítulo 3, el histograma de una imagen no tiene en cuenta la información espacial sino solamente la distribución de grises en la imagen. Por ello, dos imágenes muy diferentes pueden tener el mismo histograma. Ello hace que los métodos de segmentación basados en la búsqueda de umbrales, mediante análisis del histograma, resulten limitados en muchos problemas reales.

4.3. Técnicas basadas en la detección de bordes

La segmentación basada en *detección de bordes* agrupa a un gran número de técnicas que usan la información proporcionada por las fronteras de los objetos que aparecen en una imagen. Estos bordes marcan la localización de los puntos de la imagen donde se producen discontinuidades en los niveles de gris, en el color, en la textura, etc.

Puesto que se desea encontrar los objetos individuales presentes en una imagen, parece lógico que si se encuentran las fronteras de tales objetos con el fondo se podría segmentar los objetos de la escena general.

4.3.1 Segmentación basada en las componentes conexas

Utilizando el concepto de componente conexa, estudiado en el capítulo 2, se puede plantear el detectar los objetos presentes en una imagen sin más que encontrar las componentes conexas de la misma. Esto ocurre cuando los objetos tienen un color uniforme y distinto del fondo, lo que permite asegurar que los bordes del objeto se corresponden con los bordes de la componente conexa.

En el caso de imágenes en blanco y negro las componentes conexas suelen corresponder a los objetos directamente. Así por ejemplo en un documento escaneado los objetos de color negro suelen ser los caracteres que se desea reconocer, y un etiquetado de componentes conexas basta para encontrarlos (ver Figura 65). Esto también ocurre en el caso de imágenes a contraluz, o de imágenes umbralizadas.

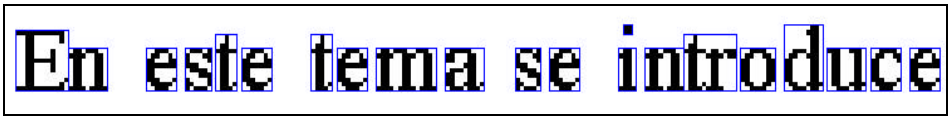


Figura 65.- Ejemplo de segmentación basada en componentes 8-conexas.

En el caso de imágenes en niveles de gris o en color, la elección del conjunto de puntos V que definen la conectividad es fundamental para que el etiquetado de componentes conexas aporte información válida sobre los objetos de una imagen.

De cualquier manera, el estudio de las componentes conexas suele ser un paso fundamental para la segmentación, pero insuficiente. Puede observarse en la Figura 65 que la *t* y la *r* forman la misma componente conexa. Por ello, posteriormente suelen aplicarse heurísticas particulares para conseguir una segmentación adecuada.

Algoritmo de etiquetado de componentes conexas

En este punto se expone un algoritmo eficiente de etiquetado de componentes conexas. Este algoritmo tendrá por entrada una imagen y devolverá una etiqueta asociada a cada píxel que indicará a qué componente conexa pertenece.

Un algoritmo de este tipo se puede construir fácilmente usando un esquema recursivo de búsqueda con retroceso. Tal algoritmo recorre la imagen de

izquierda a derecha y de arriba a abajo. Cuando encuentra un píxel a negro le asigna una etiqueta de un contador que posee y entra en una función recursiva que recorre los píxeles adyacentes, siguiendo un orden determinado, marcándolos como visitados y asignándoles el mismo valor del contador. Una vez recorridos todos los píxeles de ese objeto incrementa el contador y sigue recorriendo la imagen en busca del siguiente píxel a negro.

El algoritmo explicado no es eficiente por dos motivos:

- El uso de la recursividad no es una buena idea cuando el número de llamadas recursivas que se va a realizar es grande. Una llamada recursiva a una función implica la copia del estado del sistema y la asignación de nuevas variables, siendo en general poco eficiente en cuanto a tiempo y memoria. Realizar este tipo de operaciones por cada píxel de una imagen puede implicar varios millones de llamadas a función, lo que termina generalmente con un error por falta de memoria y en el mejor de los casos en una implementación lenta.

- Se puede llegar a preguntar varias veces por cada píxel de la imagen. Cuando se encuentra el primer píxel de un objeto se visitan los puntos de ese objeto, puntos por los que luego se va a volver a preguntar, aunque ya han sido visitados.

El primer problema se puede subsanar usando una pila y un bucle para imitar el comportamiento de la recursividad. Con este pequeño cambio se consigue aumentar la velocidad y que el algoritmo funcione incluso sobre imágenes grandes.

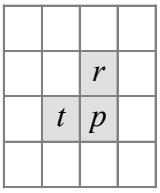


Figura 66.- Los píxeles *r* y *t* son los vecinos a tener en cuenta al evaluar el píxel *p* en el algoritmo de etiquetado de componentes conexas.

La solución del segundo problema se resuelve con una implementación diferente que es más eficiente (ver el algoritmo del cuadro adjunto). Esta implementación también recorre los píxeles de la imagen de izquierda a

derecha y de arriba abajo. Cuando encuentra un píxel activo mira si el vecino superior o el vecino anterior estaban etiquetados (ver Figura 66). En caso afirmativo, le pone la misma etiqueta que tenga alguno de estos y continúa el proceso. En caso contrario, toma una nueva etiqueta de un contador que acto seguido incrementa. Además, cuando encuentra dos píxeles activos adyacentes con diferente etiqueta anota su equivalencia en una lista.

Cuando termina de recorrer la imagen, se necesita calcular la matriz del cierre transitivo de la lista para conocer las equivalencias finales entre los píxeles que se ha ido etiquetando. Esto se puede resolver aplicando el *algoritmo de Warshall* que agrupa los píxeles en clases de equivalencia. Este algoritmo permite una optimización del cálculo tradicional de la matriz M del cierre transitivo, que consiste en el producto binario de la matriz de adyacencia por sí misma hasta que no se produzcan cambios. Luego, como representante de cada clase de equivalencia basta con tomar el mínimo valor.

Sin embargo, de nuevo, el cálculo de esta matriz puede ser muy costoso, sobre todo cuando el número de componentes conexas sea elevado, por lo que es preferible recurrir a técnicas alternativas que permitan calcular la última implicación de cada elemento de la lista antes que la matriz del cierre transitivo. Una de las formas posibles consiste en construir la lista de manera que sus elementos siempre tengan la forma $m \circledR n$ donde $m>n$. Luego basta con iterar sobre los elementos de tal lista hasta conseguir que todos los elementos de la lista impliquen al menor posible, con lo que se consigue que todas las implicaciones de la lista, que se refieran a la misma componente conexa, apunten todas al mismo elemento.

Por último, a la hora de implementar de manera óptima éste y otros algoritmos, no debe olvidarse que en última instancia las imágenes se almacenan en la memoria física en el ordenador. Así, cuando sea posible, el realizar operaciones que traten la memoria física directamente, en vez de abstracciones como píxeles y colores, suele redundar en un aumento de la eficiencia de la implementación.

- Algoritmo de Etiquetado de componentes 4 conexas -

Paso 1.- cont $\leftarrow$ 1

Paso 2.- Según la Figura 66, si $p \in V \Rightarrow$

Si $r \notin V$ y $t \notin V \Rightarrow$ Etiq(p) $\leftarrow$ cont

cont++

Si $r \notin V$ y $t \in V \Rightarrow$ Etiq(p) $\leftarrow$ Etiq(t)

Si $r \in V$ y $t \notin V \Rightarrow$ Etiq(p) $\leftarrow$ Etiq(r)

Si $r \in V$ y $t \in V$ y Etiq(t) = Etiq(r) $\Rightarrow$ Etiq(p) $\leftarrow$ Etiq(t)

Si $r \in V$ y $t \in V$ y Etiq(t) $\neq$ Etiq(r) $\Rightarrow$ Etiq(p) $\leftarrow$ Etiq(t)

$M(r,t) = M(t,r) = 1$

Paso 3.- Hacer p igual al siguiente píxel, siguiendo el orden de izquierda a derecha y de arriba a abajo. Si quedan más píxeles ir al paso 2.

Paso 4.- Calcular el cierre transitivo de la matriz M y sustituir la etiqueta de cada píxel por su equivalente de orden menor.

Ejemplo 11.-

Para etiquetar cada píxel de la Figura 67a según la componente conexa a la que pertenezcan se aplica el algoritmo 1. Al llegar al paso 4 da como resultado la imagen (b) de la Figura 67, y la siguiente matriz M.

$$M = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

El cálculo de la matriz M_T del cierre transitivo, consiste en un producto sucesivo, de M por si misma, hasta que no se producen cambios:

$$M_0 = M$$

$$M_1 = M_0 \cdot M_0 = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$M_2 = M_1 \cdot M_1 = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = M_T$$

De la matriz M_T se deduce:

$$2 \rightarrow 1,\ 3 \rightarrow 1$$

Sustituyendo los puntos afectados se obtiene la imagen (c) de la Figura 67.

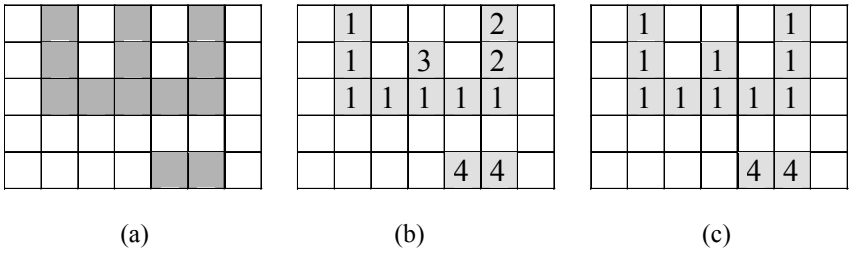


Figura 67.- La figura (a) representa un mapa de bits donde los cuadros oscuros representan píxeles negros y los cuadros blancos píxeles blancos. La figura (b) presenta los píxeles etiquetados antes del cálculo de la matriz del cierre transitivo. Finalmente (c) presenta el resultado del algoritmo de etiquetado.

4.3.2 Detección de bordes con filtros de gradiente

En el caso de que los objetos no tengan un color uniforme, o que este color pueda cambiar dependiendo del objeto y del fondo, es preciso utilizar técnicas que permitan detectar cambios entre los valores de los píxeles de las imágenes, más que fijar de antemano cuáles corresponden a objetos y cuáles no.

En el capítulo precedente se ha estudiado cómo un filtro paso alto permite la obtención de los píxeles de la imagen donde se produce un cambio brusco de intensidad. Sobre la imagen obtenida tras el filtrado, que está en niveles de gris, se puede realizar un umbralizado que separe los píxeles con valores altos del resto. Los bordes así obtenidos constituyen precisamente la frontera de los objetos con el fondo.

En general, la segmentación basada en detección de bordes presenta varios problemas. El más importante quizás consiste en la no aparición de un borde o frontera que sí exista en la imagen real. Además, junto con los bordes suele aparecer ruido que se deriva de la propia naturaleza de la imagen, esto hace que se pueda producir la aparición de un borde que no exista en la realidad. Por otra parte, tras el filtrado y el umbralizado, los bordes suelen aparecer con un grosor apreciable de varios píxeles, mientras que sería deseable que los bordes sólo tuviesen un píxel de grosor. Por último, los bordes obtenidos rara vez aparecen de forma conexa, por lo que en vez de una línea que describe el borde del objeto de manera continua se tiene una serie de fragmentos de los bordes de mayor o menor longitud. Para solucionar estos problemas aparecen enfoques y algoritmos más o menos basados en heurísticas.

Tanto el filtro de la primera derivada (basado en el operador de gradiente) como el de la segunda (basado en el operador de la laplaciana) son muy sensibles al ruido, por ello suele aplicarse previamente un filtro de suavizado para eliminarlo. El filtro del gradiente suele producir bordes gruesos, mientras que el filtro de la laplaciana, suele producir imágenes con el grosor de los bordes a un píxel. Sin embargo, el filtro de segunda derivada es más sensible al ruido que el de gradiente. Por ello suelen usarse combinados considerándose como borde aquellos píxeles donde el módulo del gradiente supera un umbral y además se produce cambio de signo en la segunda derivada, que corresponde a un paso por cero. Hay que notar que raramente coincide el valor cero con el valor de un píxel tras la aplicación de la laplaciana; ese cero se produce a resolución subpíxel y sólo es detectable por un cambio de signo en el resultado de la segunda derivada.

Finalmente, para localizar los bordes a partir de la imagen resultado del filtrado suelen aplicarse ciertos algoritmos que procesan los resultados y devuelven los segmentos que corresponden a los posibles bordes. En los siguientes puntos se presentan tres técnicas distintas que hacen esto.

Unión de segmentos mediante procesado local

Sea una imagen digital *I*. Decimos que el gradiente (∇) del píxel (*x'*, *y'*), que está en la vecindad del píxel (*x*, *y*), tiene magnitud similar a la del píxel (*x*, *y*) si y sólo si:

$$|\tilde{N}I(x, y) - \tilde{N}I(x', y')| \leq U$$

siendo *U* un umbral no negativo.

Análogamente, el píxel (*x'*, *y'*), vecino de (*x*, *y*), tiene un ángulo similar a éste si y sólo si:

$$|a(x,y) - a(x',y')| \leq A$$

siendo *A* un ángulo umbral y $a(x, y) = \text{arctg}\,(G_y/G_x)$.

Dado un punto (*x*, *y*) etiquetado como borde, su vecino es similar si los gradientes de ambos puntos son similares en magnitud y en ángulo. Un proceso iterativo que etiqueta como bordes a los píxeles similares a los de las fronteras de una imagen, y que se repite hasta que no se producen cambios, permite la obtención de una imagen de bordes en las que habrán atenuado las discontinuidades. El proceso representado en la Figura 68 permite una obtención de bordes robusta.

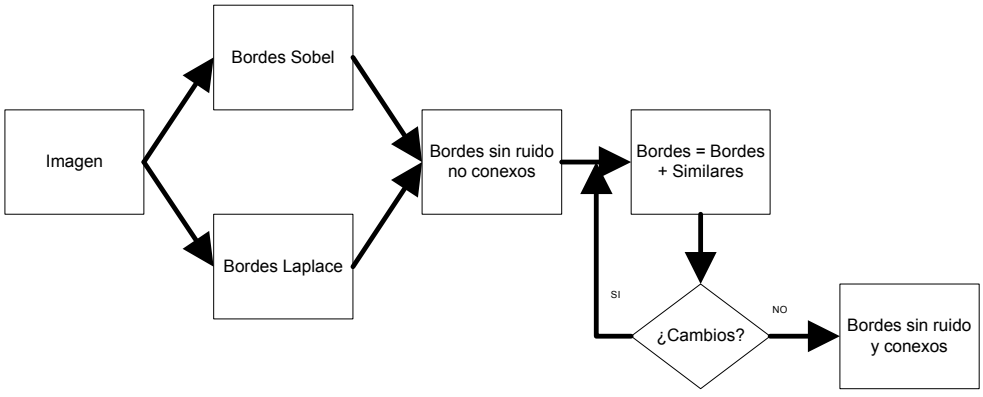


Figura 68.- Método para la obtención de bordes continuos en una imagen digital.

Unión de segmentos mediante técnicas basadas en grafos

En este enfoque se comienza por representar mediante un grafo los segmentos extraídos de una imagen tras una operación de detección de bordes con filtros de gradiente. Luego, se buscan en dicho grafo los caminos de coste mínimo que representarán las fronteras de las regiones identificadas.

Recordemos brevemente algunas definiciones sobre grafos que serán de utilidad en este apartado. Un *grafo* $G = (N, A)$ está formado por una conjunto finito y no vacío de *nodos* N, y un conjunto A de pares (n_i, n_j), ordenados o no, entre elementos de N llamados *arcos*. Un grafo en el que todos los arcos están ordenados (orientados) se llama *grafo dirigido*, y en caso contrario el grafo es *no dirigido*. Si un arco es dirigido (n_i, n_j) desde el nodo n_i hasta el nodo n_j, entonces n_i es el nodo padre o antecesor y n_j es el nodo sucesor. Si a los arcos (n_i, n_j) se les asocia un coste correspondiente $c(n_i, n_j)$, se dice que el grafo es *ponderado* o *valorado*. Un *camino* en un grafo dirigido G es una secuencia de nodos: $n_1, n_2, ..., n_k$, donde los arcos (n_1, n_2), ..., (n_{k-1}, n_k) pertenecen al conjunto de arcos. El *coste* de dicho camino c viene dado por la expresión:

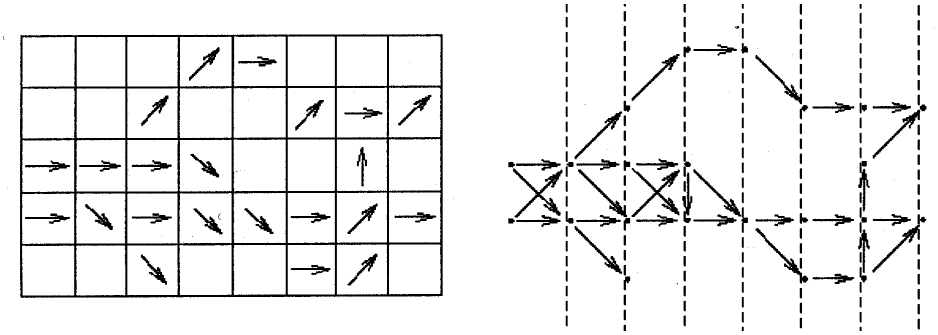
$$c = \sum c(n_{i-1}, n_i) \qquad 1 < i \leq k$$

Un *componente de borde* es la frontera entre dos píxeles p y q, tal que p y q son vecinos 4-conexos. Un *borde* o *frontera* se puede definir como una secuencia de componentes de borde. En este contexto, el *coste de un componente* de borde, definido por los píxeles p y q, viene dado por:

$$c(p, q) = H - |I(p) - I(q)| \tag{4.1}$$

siendo H en valor de intensidad más alto en la imagen (en imágenes de niveles de gris es 255), e $I(p)$ e $I(q)$ los valores de intensidad correspondientes a los píxeles p y q, respectivamente.

La Figura 69 muestra una imagen de bordes, donde se ha representado sólo las componentes significativas. A partir de ellas puede obtenerse un grafo (ponderado) construido a partir de la imagen de bordes según el procedimiento descrito. Luego sobre el grafo construido puede definirse un algoritmo de búsqueda heurística que permite obtener el camino de menor coste entre dos nodos, n_O y n_D, del grafo. El algoritmo es el siguiente:



(a) (b)

Figura 69.- La cuadrícula representa un mapa de píxeles sobre el que se superpone el grafo de los bordes: (a) direcciones de los bordes, y (b) grafo correspondiente.

- Algoritmo de búsqueda del camino de menor coste -

Paso 1.- Expandir el nodo origen n_O y poner todos sus sucesores $\{n_i\}$ en una lista L. En ella todos los nodos tiene un puntero "hacia detrás" a n_O. Evaluar la función de coste $r(n_i)$ a todo nodo expandido n_i desde n_O, que inicialmente valdrá $c(n_O, n_i)$ según la expresión (4.1).

Paso 2.- Si la lista L es vacía, acabar con fallo; en otro caso, determinar el nodo n_j de la lista L cuya función de coste asociada $r(n_j)$ sea la menor y quitar n_j de la lista L. Si $n_j = n_D$ (nodo final del camino), recorrer el camino de punteros "hacia detrás", encontrar el valor mínimo y acabar con éxito.

Paso 3.- Si la opción de parar no fue tomada en el paso 2, expandir el nodo especificado n_j y poner sus sucesores en la lista L con punteros "hacia detrás" a n_j. Calcular los costes según la función r (si n_k es un sucesor de n_j en L, su coste viene dado por el coste $r(n_j)$ para ir de n_i a n_j más el coste del arco $c(n_j, n_k)$). Volver al paso 2.

Para la aplicación del algoritmo anterior, la función de costes r debe ser separable y monótona con respecto de la longitud del camino. Además, los costes locales de componentes de bordes deben ser no negativos. Este algoritmo puede presentar problemas al poder aparecer, siguiendo el proceso descrito, un ciclo infinito en la búsqueda del camino. Este problema puede resolverse llevando una lista de nodos visitados y no expandiendo un nodo ya visitado. En general, el encontrar un camino de coste mínimo entre dos nodos puede resultar costoso computacionalmente. Normalmente, se prefiere perder algo de eficiencia con tal de tener un algoritmo más rápido.

También, basándose en un modelo de grafo puede aplicarse la técnica de *programación dinámica*. La programación dinámica se basa en el *principio de optimalidad*. Este principio, aplicado al problema de un camino entre dos nodos de un grafo (o búsqueda de una frontera entre píxeles) significa que: "si el camino optimal entre el nodo origen n_O y el nodo destino n_D pasa por un nodo intermedio n_E, entonces los subcaminos que van desde n_O y n_E, y entre n_E y n_D, respectivamente, son también optimales". El problema planteado es equivalente a encontrar un camino de coste optimal en un grafo multietapa, cuya formulación matemática puede consultarse en un libro sobre técnicas de diseño de algoritmos.

El grafo multietapa se construye a partir de la imagen de gradientes tras aplicar las reglas de conexión de bordes discontinuos. Usando estos datos se define un grafo multietapa como por ejemplo el de la Figura 70. Los costes asociados a los arcos indican la magnitud de los valores del gradiente de los bordes, mientras que las direcciones de gradiente han sido usadas para definir las reglas de conexión. Los arcos en trazo grueso de la figura señalan el camino optimal (aquél de coste máximo) que indica los bordes más probables entre un nodo origen y otro destino.

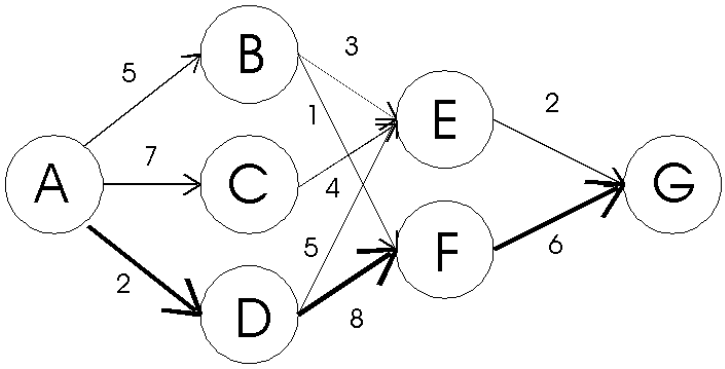


Figura 70.- Grafo 4-etapas y su camino óptimo (de coste 16).

Transformada de Hough

Al igual que las técnicas basadas en grafos, la *transformada de Hough* es un método de análisis global que se diseñó para detectar líneas rectas y curvas a partir de las posiciones de *n* puntos. Una ventaja de esta técnica es la robustez de los resultados de segmentación conseguidos al aplicarla; sin embargo, su coste computacional es elevado.

El algoritmo propuesto por Hough en 1962, conocido como transformada Hough, permite determinar el conjunto de rectas que probablemente forman una nube de puntos. Este algoritmo parte de la consideración de que para cualquier punto (x_i,y_i), el conjunto de rectas que pasan por él cumple la ecuación:

$$y_i = a\,x_i + b \tag{4.2}$$

siendo a y b, los parámetros que determinan las infinitas rectas que pasan por el punto (x_i,y_i). Para otro punto (x_j,y_j), las rectas que pasan por él siguen la ecuación:

$$y_j = a\,x_j + b \tag{4.3}$$

donde a y b son parámetros variables de nuevo. La recta que pasa a la vez por (x_i,y_i) y por (x_j,y_j) tiene como valores de los parámetros (a,b) el resultado de resolver el sistema planteado por (4.2) y (4.3), que llamaremos a' y b'.

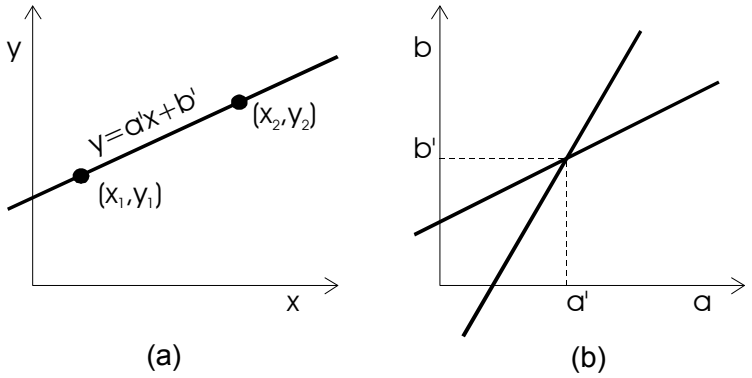


Figura 71.- Representación de la recta $y = a'x + b'$ en el espacio (x,y) y en el espacio de parámetros (a,b).

Si se representa el espacio de los parámetros (a,b), el número de veces que una recta pasa por el punto (a',b') determina el número de píxeles que comparten la misma recta. Por ello, el simple conteo de las veces que se repiten los mismos valores para a y b sirve de indicador para encontrar las rectas que existen en una imagen.

Más concretamente, si se representa el espacio de parámetros (a,b) de las rectas que pasan por los puntos (x_1,y_1) y (x_2,y_2) se obtiene la Figura 71b.

La intersección de ambas rectas determina el punto (*a'*,*b'*) que se corresponde con los parámetros de la recta *y=a'x+b'*, que contiene los puntos (x_1,y_1) y (x_2,y_2). De manera práctica, suele discretizarse el plano de los parámetros (*a*,*b*), para realizar el conteo. Además, la representación *y=ax+b* plantea el problema de que ni los valores de *a* ni los de *b* están acotados, complicándose aún más en el caso de las rectas verticales en los que el parámetro *a* tiende a infinito. Es por esto que se usa la representación en polares de una recta:

$$x_i cos\theta + y_i sen\theta = r$$

siendo θ y r los nuevos parámetros que determinan los infinitos puntos que pasan por x_i e y_i. Nótese que en esta ecuación el parámetro θ está acotado en el intervalo $[0,\pi]$.

Ejemplo 12.-

Determinar las dos rectas que con mayor probabilidad aparecen en la imagen de la Figura 72, obtenida mediante un filtrado de Sobel.

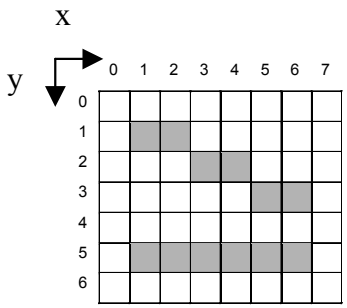


Figura 72.- La figura presenta un mapa de bits de ejemplo para la aplicación de la transformada de Hough.

Para el punto de coordenadas (1,5), las rectas que pueden encontrarse tienen los parámetros: (θ =0º, r = 1) ó (θ =10º, r = 1,16) ó (θ =20º, r = 1,28) ó …., que se deducen de la ecuación: $x_i cos\theta + y_i sen\theta = r$. Calculando el valor de r para cada θ posible (en intervalos de diez en diez grados), se obtienen los resultados que aparecen en la Tabla 4.

	q	0	10	20	30	40	50	60	70	80	90	100	110	120	130	140	150	160	170
X	***Y***																		
1	1	1	1,16	1,28	1,37	1,41	1,41	1,37	1,28	1,16	1	0,81	0,6	0,37	0,12	-0,1	-0,4	-0,6	-0,8
2	1	2	2,14	2,22	2,23	2,17	2,05	1,87	1,62	1,33	1	0,64	0,26	-0,1	-0,5	-0,9	-1,2	-1,5	-1,8
3	2	3	3,3	3,5	3,6	3,58	3,46	3,23	2,91	2,49	2	1,45	0,85	0,23	-0,4	-1	-1,6	-2,1	-2,6
4	2	4	4,29	4,44	4,46	4,35	4,1	3,73	3,25	2,66	2	1,28	0,51	-0,3	-1	-1,8	-2,5	-3,1	-3,6
5	3	5	5,44	5,72	5,83	5,76	5,51	5,1	4,53	3,82	3	2,09	1,11	0,1	-0,9	-1,9	-2,8	-3,7	-4,4
6	3	6	6,43	6,66	6,7	6,52	6,15	5,6	4,87	4	3	1,91	0,77	-0,4	-1,6	-2,7	-3,7	-4,6	-5,4
1	5	1	1,85	2,65	3,37	3,98	4,47	4,83	5,04	5,1	5	4,75	4,36	3,83	3,19	2,45	1,63	0,77	-0,1
2	5	2	2,84	3,59	4,23	4,75	5,12	5,33	5,38	5,27	5	4,58	4,01	3,33	2,54	1,68	0,77	-0,2	-1,1
3	5	3	3,82	4,53	5,1	5,51	5,76	5,83	5,72	5,44	5	4,4	3,67	2,83	1,9	0,92	-0,1	-1,1	-2,1
4	5	4	4,81	5,47	5,96	6,28	6,4	6,33	6,07	5,62	5	4,23	3,33	2,33	1,26	0,15	-1	-2	-3,1
5	5	5	5,79	6,41	6,83	7,04	7,04	6,83	6,41	5,79	5	4,06	2,99	1,83	0,62	-0,6	-1,8	-3	-4,1
6	5	6	6,78	7,35	7,7	7,81	7,69	7,33	6,75	5,97	5	3,88	2,65	1,33	-0	-1,4	-2,7	-3,9	-5

Tabla 4.- Representación del valor de r a partir de las coordenadas de un punto (x,y) y del ángulo q.

Redondeando al valor entero más próximo y agrupando las repeticiones de las parejas (q, r) se obtiene la Tabla 5. En ella se aprecia que para (q = 90º, r = 5), y para (q = 120º, r = 1), se obtienen dos valores máximos locales que se corresponden a sendas rectas que aparecen en la Figura 73.

θ/ρ	0	10	20	30	40	50	60	70	80	90	100	110	120	130	140	150	160	170
8				1	1	1												
7		1	2	2	2	1	2	1										
6	2	2	2	2	3	4	3	3	3									
5	2	2	2	1	1	1	3	4	3	6	2							
4	2	2	3	3	3	2	1		2		4	3	1					
3	2	2	1	1		1	1	2	1	2		3	2	2				
2	2	2	1	1	1	1	1	1	1	2	2		2	1	2	1		
1	2	1	1	1	1	1	1	1	2	2	4	5	1	2	1	1	1	
0												1	6	3	2	2	1	1
-1														3	4	2	2	2
-2														1	2	3	3	2
-3															1	2	2	2
-4																1	2	3
-5																	1	2
-6																		
-7																		
-8																		

Tabla 5.- Representación del conteo de los valores de parámetros considerados, que describen las rectas más probables que aparecen tras la aplicación de la transformada Hough.

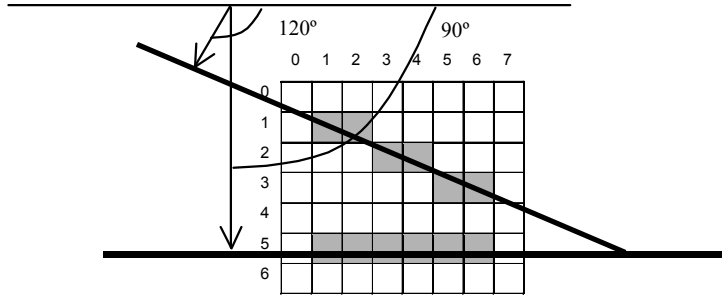


Figura 73.-Rectas obtenidas por la aplicación de la transformada de Hough representadas sobre el mapa de bits de la Figura 72.

4.4. Técnicas basadas crecimiento de regiones

Las técnicas agrupadas bajo el nombre de *crecimiento de regiones* determinan zonas dentro de una imagen basándose en criterios de similaridad y proximidad entre los píxeles de la misma. En estas técnicas la homogeneidad (o falta de homogeneidad) entre regiones adyacentes es el criterio utilizado para unir (o dividir) regiones de la imagen. Dicha homogeneidad se puede definir a partir de criterios como: el nivel de gris medio, el color, la forma, etc. El resultado de la segmentación es una partición de la imagen en regiones homogéneas.

Las técnicas de segmentación basadas en bordes tratan de encontrar fronteras entre regiones. En general, las técnicas basadas en regiones trabajan mejor en imágenes con ruido, en donde los bordes son difíciles de localizar. La segmentación resultante de una detección de bordes y la basada en crecimiento de regiones, aplicadas a una misma imagen no producen normalmente el mismo resultado. Es posible combinar los resultados producidos por ambos tipos de segmentaciones.

4.4.1 Unión de regiones

Este procedimiento agrupa píxeles de la imagen formando regiones de similares características. Inicialmente se elige una colección de píxeles de manera aleatoria que actúan como “semillas” para comenzar el crecimiento. A estos puntos de la

imagen se les agregan otros adyacentes si tienen valores que cumplen cierto criterio definible de homogeneidad con los de los puntos semilla. Si ocurre esto, pertenecen a la misma región y pasan a tener los mismos valores que los puntos semilla. El criterio de agregación más usual suele consistir en que un píxel, adyacente a una región, se agregue a ésta si su intensidad es similar a la media de las intensidades de los píxeles de la región. El principal inconveniente del método se deriva de que el resultado depende de la elección inicial de los puntos semilla.

En la Figura 74 se presenta un ejemplo en el que se ha usado unión de regiones para segmentar objetos. La figura corresponde a una fotografía de dos cartulinas circulares de colores sobre un tablero negro. El objetivo es localizar las dos cartulinas. El criterio de agregación en este caso consistió en la similitud de la terna RGB que posee cada píxel.

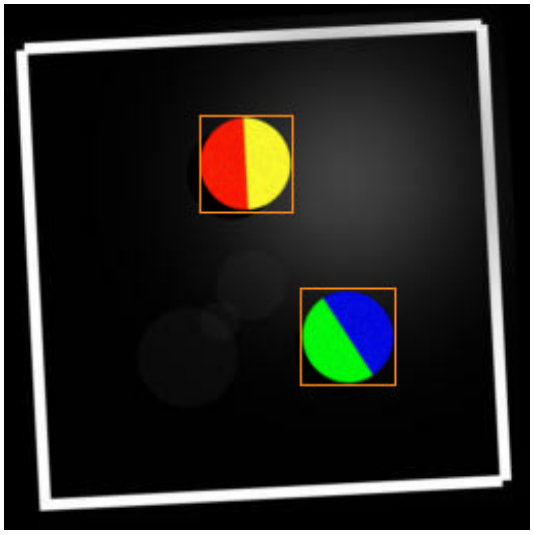


Figura 74.- Segmentación de objetos por unión de regiones. Las regiones correspondientes a los colores amarillo, rojo azul y verde crecen por agregación de píxeles con matiz similar.

4.4.2 División de regiones

Es un proceso en cierta forma opuesto al de unión de regiones. Se parte una única región que representa a toda la imagen, y si dicha región no satisface el criterio de homogeneidad establecido, la región inicial se divide, de manera secuencial, en subregiones de las que se estudia su homogeneidad. Si una subregión está formada por puntos homogéneos, no se subdivide; si no, se sigue dividiendo. Los criterios

utilizados para dividir regiones son similares a los usados para agruparlas, y sólo se diferencian en la dirección en que se aplican.

4.4.3 División y unión de regiones (”*split and merge*”)

Horowitz y Pavlidis propusieron en 1976 un método que solventa el problema de la elección arbitraria de “semillas” para agrupar regiones. El algoritmo recibe la imagen a segmentar y devuelve una imagen resultado formada por regiones homogéneas. El algoritmo propuesto sigue la técnica de “divide y vencerás”. El algoritmo tiene dos fases: dividir la imagen en subimágenes o regiones (*split*) y luego agrupar subimágenes similares (*merge*).

Detallando el algoritmo, primero se define una medida que determina la similitud entre regiones de la imagen. Por ejemplo, se podrían utilizar los valores de la media y la desviación típica de los niveles de gris de los píxeles de la región. Una vez definida, dicha medida se aplica inicialmente a toda la imagen. Si esta medida supera un cierto umbral, la imagen inicial se divide en cuatro subimágenes de igual tamaño (cuadrantes). Después, se vuelve a calcular esta medida sobre cada una de las cuatro subimágenes. Si la medida en alguna de las imágenes resultantes de la división vuelve a sobrepasar de nuevo el umbral, ésta se vuelve a subdividir en cuatro partes iguales, y así sucesivamente. Cuando la medida sobre alguna de las regiones no supera el umbral, se la etiqueta como indivisible con el valor calculado. En caso contrario, la subdivisión continúa hasta que el tamaño de la región considerada sea de un píxel, momento en que dicha región se etiqueta igualmente con el valor de la medida correspondiente.

La segunda fase del algoritmo consiste en unir pares de regiones adyacentes que presenten un valor similar de la medida (de acuerdo a un nuevo umbral). El procedimiento se itera hasta que no quedan zonas adyacentes con similares características que sigan separadas. La aplicación del criterio de unión mediante fuerza bruta puede ser costoso en imágenes grandes. Con objeto de minimizar el número de comparaciones puede usarse una estructura de tipo *quadtree* para almacenar la representación de las regiones (ver figura adjunta). Esta representación permite aplicar reglas como la de que regiones de tipo *A* sólo adyacen a regiones de tipo *B* y *C*. Con estas reglas el número de posibles comparaciones se reduce.

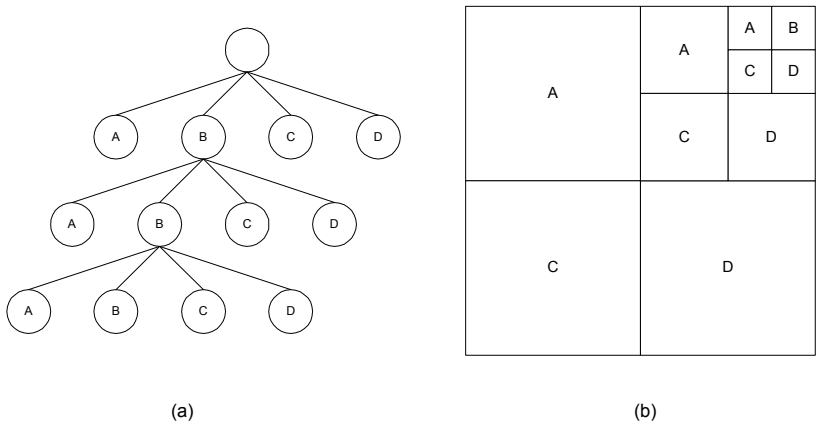
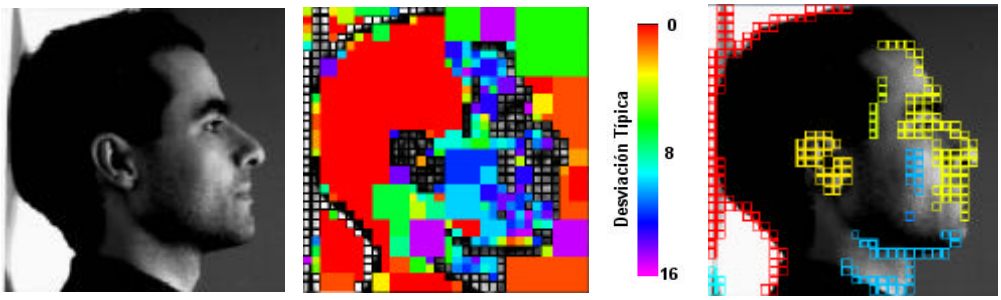


Figura 75.- Representación en quadtree (b) de las regiones de la imagen (a).

Horowitz y Pavlidis utilizaron como criterio de referencia la media de los valores de intensidad de los píxeles de la región, aunque el criterio de referencia puede ser cualquier otro.

La Figura 76 muestra la aplicación de este método de segmentación para la localización de la oreja en una imagen facial tomada de perfil. El criterio adoptado asume que las zonas con alta desviación típica corresponden a la oreja y a los bordes del rostro con el fondo. En esta implementación no se ha continuado la división de regiones no homogéneas hasta el límite de un píxel, sino que si la región a evaluar era menor de 10×10 píxeles y su desviación típica superaba el umbral fijado, dicha región se etiquetaba como una zona candidata a contener la oreja. En la Figura 76 (a) se presenta la imagen inicial, en (b) se aprecia las regiones resultantes de aplicar el algoritmo de división y unión de regiones propuesto, (c) muestra las regiones independientes de alta varianza que aparecen en la imagen. Finalmente la región seleccionada para contener la oreja debe cumplir ciertos criterios heurísticos. En este ejemplo, un criterio heurístico final para extraer la oreja consistió en establecer que la región que la contuviese debería cumplir unas proporciones determinadas (ancho/alto≈0,625). Esto se hizo porque se observó que de las regiones de alta varianza obtenidas aplicando el procedimiento de división y mezcla, sólo la región que contiene la oreja cumple estas proporciones.



(a) (b) (c)

Figura 76.- Segmentación de objetos mediante el algoritmo Split and Merge. Aplicación de la segmentación a la extracción de la oreja en una imagen de perfil.

4.4.4 Segmentación basada en morfología: "*watershed*"

La técnica de *watershed* (en español: "línea de división de aguas") es un procedimiento de segmentación, basado en morfología matemática, que permite extraer las fronteras de las regiones que hay en una imagen. Esta técnica aplicada en el procesamiento de imágenes fue introducida por C. Latuejéoul como *transformación de watershed* y más tarde mejorada juntamente con S. Beucher, y denominada watershed. Combinada con otras herramientas morfológicas genera unos resultados de segmentación bastante aceptables.

Inicialmente, la imagen puede verse como una representación topográfica de un terreno, donde a cada píxel se le asocia como valor de "altura" su nivel de gris correspondiente. A continuación, imaginemos que se comienza a inundar esta superficie topográfica desde los niveles más bajos de altura (valores mínimos locales, que constituyen cuencas de inundación). Llega un momento en el que, siguiendo este proceso de inundación, las aguas de cuencas contiguas se unen. Las líneas de unión, que representan las fronteras de regiones homogéneas, constituyen el resultado de la segmentación. Por lo tanto, esta transformación de watershed particiona una imagen en niveles de gris en otra formada por regiones significativas, separadas por las líneas resultantes mencionadas. El proceso se ilustra en la Figura 77.

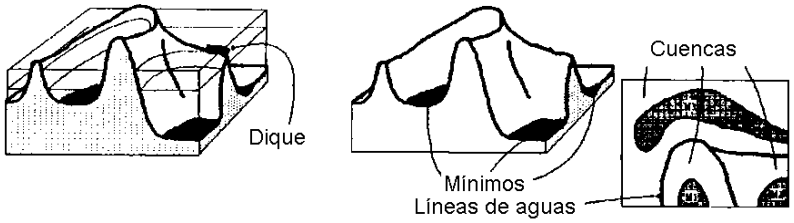


Figura 77.- Una imagen que ilustra el símil del watershed en una superficie de un terreno.

La obtención de las líneas de división de aguas de una imagen digital no es tarea sencilla. Este tipo de algoritmos suelen ser lentos (y, a veces, poco precisos). Se ha investigado y desarrollado diferentes versiones del watershed para hacerlo más eficiente y rápido. El uso de colas ordenadas por niveles de gris permite mejorar la eficiencia de esta técnica de segmentación.

En este tipo de algoritmos es necesario implementar un paso previo a la segmentación, del cual dependerá el buen resultado del proceso genérico. En el caso de las imágenes en niveles de grises, el concepto de zona contenida dentro de un contorno se refiere a todos los puntos que aún estando cercanos no existe entre ellos un salto radical en la escala de gris. Basándose en este punto de partida, el algoritmo comenzará con la detección de unos *marcadores básicos*. Estos marcadores se escogerán entre los mínimos locales de la imagen. A partir de éstos se empieza el proceso de inundación. Tras este paso, cada objeto que aparece en la imagen deberá tener un marcador diferente asociado.

Está técnica de segmentación está aconsejada para imágenes con texturas homogéneas y con gradientes de intensidad débiles. Finalmente, los objetos resultantes de la segmentación se corresponden con los mínimos del gradiente morfológico y con los contornos de las líneas de división de aguas del gradiente.

Uno de los problemas de este algoritmo es el de la sobresegmentación. Este problema se caracteriza por una detección de un número de regiones excesivo, muchas de las cuales no son importantes dentro de la imagen, o no representan objetos existentes en la imagen original. El origen del problema suele estar en que la elección de marcadores no ha sido adecuada, por ejemplo se ha seleccionado marcadores que eran producidos por ruido en la imagen original. Es por ello que previo a este algoritmo suele aplicarse un filtrado de suavizado para eliminar marcadores no significativos.

Para finalizar este apartado se señalan algunos criterios a tener en cuenta para obtener un buen resultado en este proceso:

- Filtrar las imágenes para dejar sólo las zonas importantes de ella.

- Elegir unos marcadores adecuados. Dependiendo del problema que se trate, se tendrá que definir una heurística diferente para elegir los marcadores con los que se tendrá que empezar el proceso.

4.5. Otros enfoques para la segmentación

Actualmente, la segmentación de imágenes continúa siendo un área activa de investigación. Existen numerosas técnicas de segmentación, aparte de las explicadas hasta ahora, que no pueden ser englobadas estrictamente en ninguno de los tres grupos descritos. Por ello, en los siguientes puntos, se repasa, brevemente y de manera separada, las técnicas basadas en el uso del color, en el análisis de la textura y en el movimiento (obtenido como una sucesión de imágenes consecutivas en el tiempo).

4.5.1 Segmentación basada en el color

Durante mucho tiempo, la segmentación se realizó solamente usando imágenes de niveles de gris. Con el aumento de las prestaciones de los computadores usados en los sistemas de visión es posible realizar segmentación de imágenes en color, aprovechando la información que el color aporta.

Ya se ha estudiado que el color se puede representar como la unión de tres planos, cada uno con la información relativa a la intensidad de cada punto respecto a cada una de la componentes de una base de color (rojo, verde y azul en el modelo RGB). Las técnicas de segmentación basadas en el color, consisten básicamente en la aplicación de las mismas técnicas usadas en imágenes de grises sobre cada uno de los planos de color por separado. En una segunda etapa, se combinan usando ciertos criterios los resultados de la segmentación obtenidos en cada plano para producir como resultado la segmentación de la imagen en color.

Una técnica de este tipo se basa en la aplicación de crecimiento de regiones. En concreto, se aplica una variante que contempla el color del algoritmo

de “split-and-merge”, descrito al explicar la segmentación basada en regiones. El algoritmo queda de la siguiente forma:

- Algoritmo Split and Merge para segmentación basada en el color-

Paso 1.- Se calculan las características de color usando los valores de las componentes de los planos rojo, verde y azul de una imagen RGB.

Paso 2.- La imagen se divide en regiones cuadradas de igual tamaño, usando la estructura de datos de árbol cuaternario o “*quadtree*”.

Paso 3.- Cuatro cuadrantes situados a un mismo nivel de subdivisión son mezclados si se satisface un cierto criterio de homogeneidad. Un cuadrante se subdivide en otros cuatro si no se satisface una condición de homogeneidad.

A continuación, se enumeran otras dos técnicas descritas en la literatura sobre la segmentación basada en el color:

- Umbralización global de la componente de color usando el espacio de representación HSV (Hue-Saturation-Value).

- Uso de algoritmos de agrupamiento o *clustering* (ver capítulo 5 de estos apuntes); en particular, la aplicación del algoritmo de las *k-medias* sobre los píxeles de la imagen en color.

La Figura 78 muestra un ejemplo de segmentación de imagen en color.

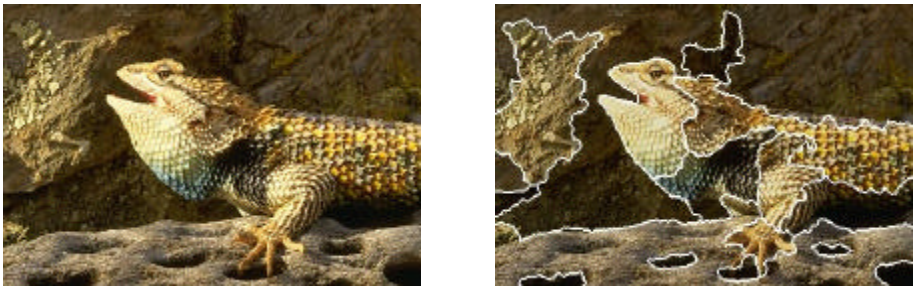


Figura 78.- Ejemplo de segmentación de una imagen en color.

4.5.2 Segmentación basada en la textura

En este enfoque se definen *modelos de texturas* para pensar en una imagen no como en una colección de píxeles, sino para tratarla como una función $I(x,y)$. El propósito del modelo es transformar una ventana de una imagen en una colección de valores que constituyen un *vector de características*. Este vector será un punto de un espacio *n*-dimensional. La representación será "buena" si ventanas tomadas de la misma muestra de textura están "cercanas" en el espacio de características, y si ventanas de la imagen con diferentes patrones de texturas quedan "alejadas" en el espacio de características considerado. Los modelos de textura se dividen, a grandes rasgos, en tres categorías: basados en *estructuras piramidales*, que tratan de capturar las frecuencias espaciales a distintos niveles de resolución; basados en *campos aleatorios*[20], que asumen que los valores de un píxel son seleccionados mediante un proceso estocástico bidimensional; y los basados en métodos estadísticos, que utilizan matrices de co-ocurrencias construidas a partir de las imágenes y de estas matrices se extraen una serie de medidas como la media, varianza, entropía, energía y correlación entre píxeles. Una descripción detallada de los tres modelos de texturas queda fuera de estos apuntes, remitimos nuevamente al lector a la bibliografía del capítulo.

Una vez caracterizada la textura de una imagen, mediante los modelos explicados, hay que aplicar un algoritmo de segmentación que podrá ser *supervisado* o *no supervisado*. La diferencia entre ambos enfoques radica en un conocimiento a priori o no de la tarea específica que el algoritmo lleva a cabo (en otras palabras, en el supervisado se conocen de antemano los tipos de texturas presentes y en el no supervisado, no se conocen).

Si se asume que el número de texturas diferentes presentes en la imagen es pequeño y que todas las texturas son distintas unas de otras, entonces es posible describir regiones pequeñas de textura homogénea, extraer vectores de características usando los modelos mencionados, y usar estos vectores como representantes de clases en el espacio de características. Ahora, todos los demás vectores de textura se pueden etiquetar asociándose al representante de clase más cercano. Se pueden usar redes neuronales u otro tipo de algoritmos, para ajustar mejor el sistema al modelo. Se está, en este caso, usando un *método de segmentación supervisado*.

[20] Random fields en inglés

Si sucede que el número de texturas posibles es muy grande y no se pueden realizar suposiciones sobre los tipos de texturas presentes en la imagen, se puede recurrir a usar *métodos de segmentación no supervisados*. Ahora se necesita realizar un análisis estadístico sobre la distribución de vectores de características. El objetivo es reconocer *clusters* o agrupaciones de vectores en la distribución y asignar la misma etiqueta a los componentes de cada uno de ellos. En general, estos métodos no supervisados son más difíciles de realizar. Ambos tipos de métodos requieren de una medida de distancia entre vectores de características; cuando los componentes son homogéneas puede usarse la distancia euclídea, y en otros casos funciones de distancia más complejas o incluso heurísticas basadas en experimentación.

Una vez segmentada la imagen se puede evaluar el resultado conseguido. En general, el resultado deberá ser una partición de la imagen en un número reducido de regiones, de tamaño grande y más o menos convexas. La siguiente imagen, muestra un ejemplo de imagen texturada y de cómo resultaría su segmentación.

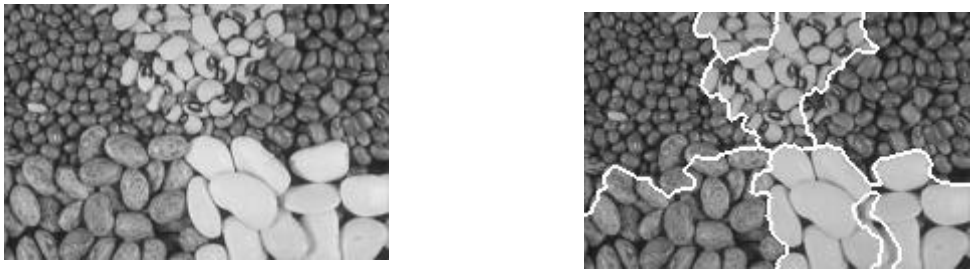


Figura 79.- (a) Imagen inicial, y (b) resultado de su segmentación.

4.5.3 Segmentación basada en el movimiento

El movimiento puede constituir una potente herramienta para la segmentación de objetos animados sobre fondos estáticos. Las técnicas básicas consisten en el estudio de la imagen resultante de la resta de dos imágenes consecutivas de una secuencia animada. Los objetos que se desplazan entre estas dos imágenes producen en la imagen resta un conjunto de píxeles con valores distintos a cero. Mientras, los elementos estáticos de la imagen, por no variar, producen cero tras la resta.

Así, partiendo de dos imágenes I_t e I_{t+1} de dos instantes consecutivos los objetos tras esta segmentación serían los píxeles a uno en la imagen I_d. Siendo U un valor umbral que depende de la variación de la iluminación entre los instantes t y t+1.

$$d_{t,t+1}(x,y) = \begin{cases} 1 & \text{si } |I_t(x,y) - I_t(x,y)| > 0 \\ 0 & \text{en otro caso} \end{cases}$$

4.6. Representación de objetos segmentados

Una vez segmentada una imagen es importante considerar la forma de describir los objetos que aparecen tras realizar los procesos de segmentación. Usando descripciones apropiadas puede conseguirse una representación única independiente de la posición, la orientación y el tamaño del objeto descrito.

4.6.1 Descripción basada en el código de cadena

Para representar el contorno de un objeto suele usarse una representación conocida como *código de cadena*. Para obtenerlo se parte de un píxel cualquiera del contorno, encadenando la dirección en que se encuentran los puntos adyacentes del contorno mediante un convenio similar al de la figura adjunta. Se consigue así una cadena de símbolos que determinan unívocamente al objeto.

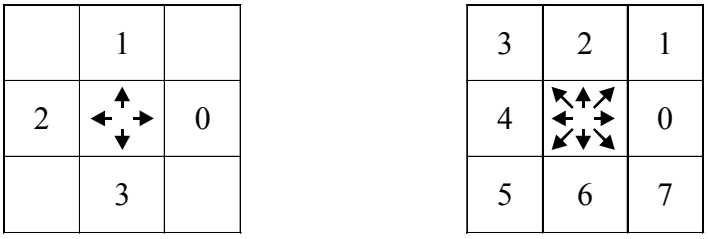


Figura 80.- Direcciones del código de cadena para 4 y 8 vecinos.

Con el fin de lograr invarianza frente a rotaciones se codifican las diferencias entre los dígitos del código y no el código mismo. Esta diferencia se realiza en módulo ocho o en módulo cuatro dependiendo del caso, no debiendo

olvidar codificar también la diferencia entre el primer y el último dígito de la cadena.

Dependiendo del punto en el que se comience el código de cadena puede ser diferente. Así, dos códigos del mismo objeto pueden tener los mismos dígitos aunque los dígitos de uno parecerán desplazados respecto a los del otro. Para lograr invarianza respecto al punto de inicio los dígitos del código obtenido, se deben disponer de forma que si se lee el código se obtenga el menor entero posible con esa secuencia de dígitos. Para ello se desplazan los dígitos a la izquierda o a la derecha hasta obtenerlo, teniendo la precaución de que los números que salgan por la izquierda entren por la derecha.

Por último, para evitar que pequeños cambios en el contorno de un objeto produzca códigos muy diferentes suele usarse algún filtrado de suavizado o algún cambio de escala sobre el objeto.

Ejemplo 13.-

En este ejemplo se desea codificar en código de cadena para 4 vecinos el objeto de la Figura 81 (a). Para ello previamente se reduce la resolución a la cuadrícula que se presenta en la Figura 81 (b).

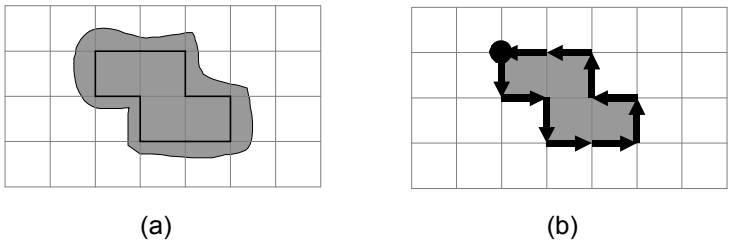


Figura 81.- Recorrido del objeto usando código de cadena de conectividad 4. El punto grueso indica el nodo de inicio.

Inicialmente se obtiene: 3030012122, que desplazándolo para obtener el mínimo queda: 0012122303. Finalmente, codificando las diferencias se obtiene: 0113101131.

4.6.2 Descripción basada en los Momentos

Dada una función $f(x,y)$ continua y acotada, se define el *momento general* de orden $p+q$ como la siguiente integral doble:

$$m_{pq} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x^p y^q f(x,y)dxdy \qquad p,q = 0,1,....\infty$$

Se puede ver que, para una función acotada en el plano, existen infinitos momentos generales obtenidos haciendo variar p y q de cero a infinito.

Se puede demostrar que dado una función $f(x,y)$ existe un único conjunto de momentos generales que la definen y viceversa.

$$f(x,y) \leftrightarrow \{m_{pq}\} \qquad p,q = 0,1,....\infty$$

En la práctica se comprueba que una cantidad menor de momentos puede describir cualquier función $f(x,y)$ con suficiente precisión. La determinación del número de momentos necesarios es particular a cada caso de estudio.

Particularizando para el caso de imágenes digitales los momentos toman la forma:

$$m_{pq} = \sum_{x=0}^{N-1}\sum_{y=0}^{N-1} x^p y^q I_D(x,y) \qquad p,q = 0,1,....\infty$$

siendo $I_D(x,y)$ una función discreta que toma valor 1 cuando el píxel pertenece al objeto y 0 cuando pertenece al fondo.

Momentos de orden cero y orden uno

El momento de orden cero (p=q=0) coincide con el área del objeto descrito.

$$m_{00} = \sum_{x=0}^{N-1}\sum_{y=0}^{N-1} I_D(x,y)$$

Los momentos de orden uno (p=0, q=1 y p=1, q=0), junto con el de orden cero, determinan el centro de gravedad de los objetos.

$$m_{10} = \sum_{x=0}^{N-1}\sum_{y=0}^{N-1} xI_D(x,y) \qquad m_{01} = \sum_{x=0}^{N-1}\sum_{y=0}^{N-1} yI_D(x,y)$$

Invarianza a traslaciones

Los momentos generales se pueden hacer invariantes a las traslaciones. Para ello basta con referirlos al centro de gravedad del objeto, es decir a los momentos de orden cero y uno. Estos momentos, que se conocen como *momentos centrales*, tienen la siguiente forma:

$$mc_{pq} = \sum_{x=0}^{N-1}\sum_{y=0}^{N-1} (x-\bar{x})^p (y-\bar{y})^q I_D(x,y) \qquad p,q = 0,1,....\infty$$

De manera más simple puede escribirse la expresión equivalente:

$$mc_{pq} = \sum_{r=0}^{p}\sum_{s=0}^{q} \binom{p}{r}\binom{q}{s} (-\bar{x})^r (-\bar{y})^s m_{p-r,q-s} \qquad p,q = 0,1,....\infty$$

Invarianza a giros

La invarianza a giros se consigue tomando una dirección de referencia para cada objeto. La dirección que se suele tomar es la que marca el *eje de mínima inercia* del objeto. Este eje se calcula teniendo en cuenta la definición de inercia:

$$I = \sum_x \sum_y \left[(x-a)sen\theta - (y-b)\cos\theta\right]^2 I_D(x,y)$$

donde (a,b) corresponden a un punto de la recta y θ al ángulo formado con la horizontal.

El eje de mínima inercia será aquél que haga cero las derivadas parciales respecto a cada variable:

$$\frac{dI}{dx} = 0 \qquad \frac{dI}{dy} = 0 \qquad \frac{dI}{d\theta} = 0$$

Desarrollando las derivadas se obtiene:

$$a = \bar{x} \qquad b = \bar{y} \qquad q = \frac{1}{2} arctg\left(\frac{2mc_{11}}{mc_{20} - mc_{02}} \right)$$

El giro del objeto y el posterior cálculo de los momentos centrales puede resumirse en la siguiente ecuación:

$$mcg_{pq} = \sum_{r=0}^{p} \sum_{s=0}^{q} (-1)^{q-s} \binom{p}{r} \binom{q}{s} (\cos q)^{p-r+s} (sen q)^{q-s+r} mc_{p-r+q-s,r+s}$$
$$p, q = 0,1,....\infty$$

Invarianza a homotecias o cambios de escala

Para evitar que la escala del objeto influya en su descripción se puede dividir los momentos por el área elevada a un factor dependiente del momento que se calcule.

$$mcgh_{pq} = \frac{mcg_{pq}}{m_{00}^{g}} \quad \text{donde } g = \frac{p+q}{2} \quad \text{y} \quad p, q = 0,1,....\infty$$

4.6.3 Descripción basada en la transformada de Fourier

Pasando del dominio de la cadena al dominio de la frecuencia se puede tratar el problema de la invarianza a las transformaciones geométricas desde un punto de vista que ofrece mejores resultados. En este punto se verá la utilidad de aplicar la transformada de Fourier sobre el código de cadena de un objeto.

Invarianza a traslaciones

Las componentes de Fourier de dos objetos iguales, que se hallen desplazados uno respecto a otro, sólo se diferencian en la componente de frecuencia cero ($F(0)$). Por tanto, la eliminación de esta componente proporciona una descripción invariante a traslaciones.

Invarianza a giros

La transformada de Fourier de un objeto girado q radianes respecto a otro igual pero que no está girado se diferencia en un factor multiplicativo e^{jq}. Por ello, suele usarse sólo los módulos que no cambian si el objeto está girado.

Invarianza a homotecias o cambios de escala

La transformada de Fourier de una secuencia de valores respecto a la de una secuencia igual pero con unos valores proporcionalmente diferentes se diferencia en que todos los elementos de la transformada han sido multiplicados por un valor k que depende del cambio de tamaño. Por ello si se dividen todas las componentes por una de ellas (normalmente suele tomarse la primera no nula) se obtiene una representación invariante a homotecias en la intensidad.

La invarianza a homotecias de tamaño es más difícil de conseguir. En este caso las frecuencias de aparición de las componentes varían proporcionalmente. Para eliminar esta variabilidad deberían dividirse todas las frecuencias por la primera frecuencia distinta de cero. Sin embargo este enfoque puede ser muy sensible al ruido presente en el código de cadena.

4.7. Conclusiones al capítulo

A lo largo del capítulo se ha introducido multitud de conceptos útiles a la hora de generar características descriptoras de los objetos. Sin embargo, cada uno de estos enfoques por separado suele resultar insuficiente para describir los objetos de la mayoría de los problemas reales. Por ello, suele usarse combinaciones de varios de los métodos propuestos y también suele realizarse modificaciones para ajustar estos métodos al problema particular que se trate.

4.8. Bibliografía del capítulo

[GW93] caps. 7 y 8,

[SHB99] cap. 5

Capítulo 5 Introducción a los clasificadores

En este capítulo se estudiarán diferentes algoritmos que permiten clasificar los elementos que aparecen dentro de una escena para poder entenderla. Los algoritmos de clasificación tienen la misión de distinguir entre objetos diferentes de un conjunto predefinido llamado *universo de trabajo*. Normalmente, el universo de trabajo se considera dividido en una colección K de *clases* (a_1, a_2... a_K), perteneciendo los diferentes tipos de objetos a algunas de estas clases.

En este capítulo se estudiarán diferentes métodos que permiten determinar, de manera automática, en qué clase se encuentra un objeto de un universo de trabajo. Estos métodos se conocen como *clasificadores*.

5.1. Características discriminantes

Para poder realizar el reconocimiento automático de los objetos se realiza una transformación que convierte un objeto del universo de trabajo en un vector[21] X cuyas N componentes se llaman *características discriminantes* o *rasgos*.

[21] En el resto del capítulo se usarán las mayúsculas para los vectores.

Estas características deben permitir discriminar a qué clases puede pertenecer cualquier objeto del universo de trabajo.

$$X = (x_1, x_2, \ldots x_N) \quad \text{con} \quad N \in \mathbb{N} \quad \text{y} \quad x_i \in \Re \quad \forall i = 1 \ldots N$$

El valor del vector de características para un objeto concreto se conoce como *patrón*. Es decir, un patrón es una instancia particular de un vector de características determinado.

La determinación de las N características discriminantes es un proceso difícil que puede no estar exento del uso de la imaginación. En general, suelen usarse características como los momentos de los objetos a reconocer, alguna transformación de los mismos (Fourier, cosenos...), las propias imágenes, o cualquier característica que se pueda obtener de los objetos mediante algún procedimiento algorítmico.

Una vez determinadas las características discriminantes para un problema concreto, la clasificación de un objeto comienza por la obtención de su patrón. El siguiente paso consiste en determinar la proximidad o grado de pertenencia de este patrón a cada una de las clases existentes, asignando el objeto a aquellas clases con las que el grado de semejanza sea mayor. A este efecto se definen las *funciones discriminantes* o *funciones de decisión* como aquellas funciones que asignan grados de semejanza de patrón a cada una de las diferentes clases.

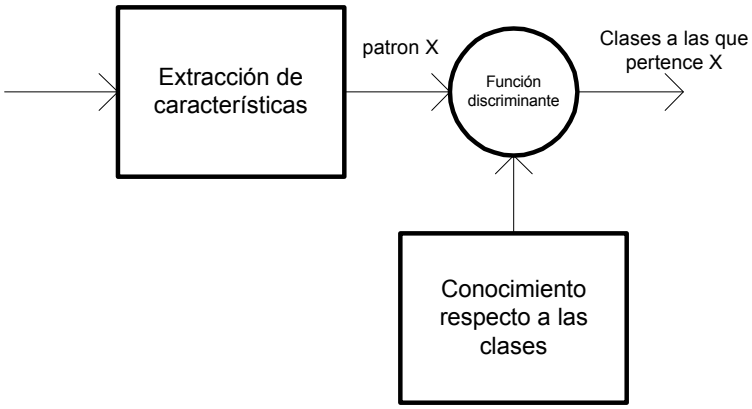


Figura 82.- Esquema general de funcionamiento de un clasificador.

Ejemplo 14.-

Supóngase una cinta transportadora transparente por la que circulan tornillos, arandelas y tuercas. Se precisa desarrollar un sistema que cuente cuántas unidades de cada tipo hay en cada momento en un intervalo de la cinta. Se supone que la iluminación se realiza a contraluz con lo que se obtienen siluetas similares a las de la Figura 83.

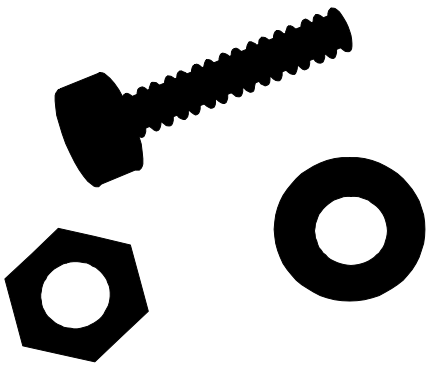


Figura 83.- Muestra de las diferentes piezas entre las que se desea distinguir, obtenidas con una cámara e iluminación a contraluz.

El primer problema, previo al reconocimiento, consiste en la segmentación de los objetos. Este paso es necesario para realizar la extracción de características de cada uno de los objetos. En este caso particular la segmentación puede reducirse al estudio de las componentes conexas, aunque en general es preciso aplicar una o varias de las técnicas de segmentación estudiadas en el capítulo anterior, hasta conseguir separar los objetos de una escena.

El segundo problema consiste en la determinación de aquellas características de los objetos que van a permitir su reconocimiento. En este caso se propone usar como características el número de agujeros presentes en la figura (uno en la tuerca y en la arandela y ninguno en el tornillo) y la desviación típica de las distancias del perímetro al centro del objeto (siempre cerca de cero en la arandela y con un valor mayor a cero en los tornillos y en las tuercas).

La Figura 84 presenta los valores de estas características para una muestra aleatoria obtenida durante 10 minutos de operación del sistema. En este gráfico se aprecia que las características seleccionadas separan adecuadamente las tres clases diferentes de objetos.

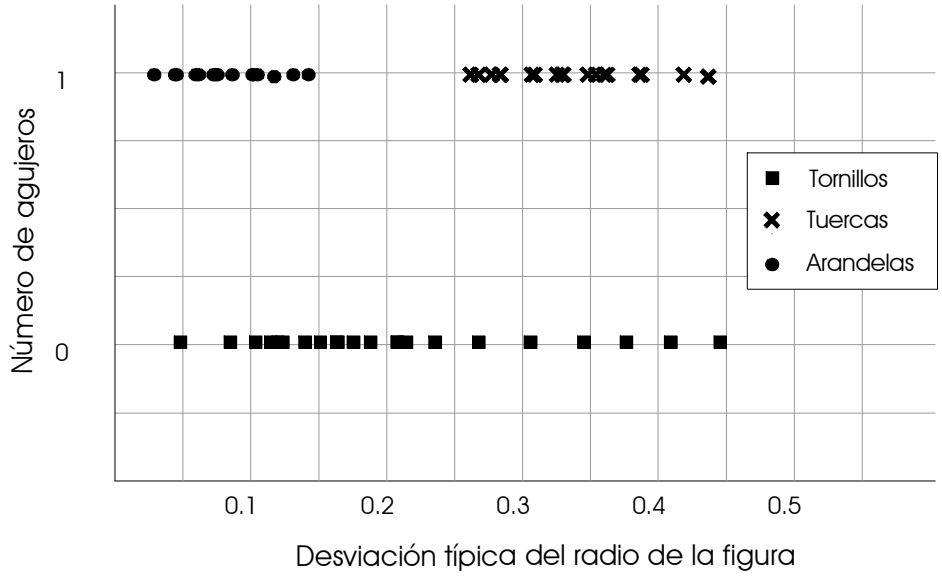


Figura 84.- Uso de dos características discriminantes (píxeles de la figura y píxeles del rectángulo que la inscribe) para distinguir entre diferentes tipos de elementos de ferretería.

5.1.1 La muestra de aprendizaje

Para poder realizar el cálculo de las funciones discriminantes suele ser precisa la existencia de un conjunto de patrones similares a los que se desea reconocer, que se denomina *conjunto de aprendizaje*. Los patrones de este conjunto se utilizan a modo de modelo para encontrar la función discriminante que clasificará correctamente los patrones del universo de trabajo. El conjunto de aprendizaje debe ser por tanto un subconjunto representativo del universo de trabajo.

Cuando la muestra es abundante suele crearse otro conjunto con ella. Este segundo conjunto se utiliza para probar los resultados de las funciones discriminantes calculadas, y se conoce como *conjunto de test*. Es importante que estos dos conjuntos sean independientes, como norma general, en el caso de un universo de trabajo grande, es suficiente con asegurar que el conjunto de aprendizaje y el test no tengan elementos en común. De esta forma puede probarse que el clasificador desarrollado ha adquirido la propiedad de *generalización*. Esta propiedad garantiza que un sistema clasifica correctamente patrones que no ha visto durante el proceso de cálculo de funciones discriminantes.

Si una vez construidos los clasificadores se prueban usando el conjunto de test y se obtienen unos resultados deficientes, deben descartarse ambos conjuntos y volver a comenzar de nuevo con nuevos conjuntos. Si para mejorar los resultados de test se volviese a realizar otro cálculo de las funciones discriminantes se estaría usando el conjunto de test de manera indirecta en el proceso de aprendizaje. Para evitar esto, cuando hay suficiente muestra, suele crearse un tercer conjunto independiente de los otros dos, llamado *conjunto de validación*. Este conjunto se utiliza para probar el sistema mientras se está construyendo, por lo que éste sí se utiliza en el proceso de cálculo de las funciones discriminantes. El objetivo de este tercer conjunto es el de impedir que el conjunto de test se use más de una vez.

Hay autores que recomiendan tomar alrededor del 60% de la muestra para construir el conjunto de aprendizaje, un 30% para el conjunto de test y el 10% restante para el conjunto de validación.

Validación cruzada

Cuando la muestra es escasa, puede no ser factible utilizar el 40% de la misma para realizar las pruebas, ya que en tal caso, no se dispondría del suficiente número de ejemplos para poder calcular las funciones discriminantes de manera correcta. Como utilizar el mismo conjunto para el aprendizaje y para el test ya se ha dicho que no es aconsejable, se suele usar un procedimiento que, si bien es mucho más lento, permite obtener una prueba estadísticamente significativa y además permite usar toda la muestra para calcular las funciones discriminantes.

El procedimiento, que se conoce como *validación cruzada*[22], consiste en el cálculo las funciones discriminantes del clasificador utilizando toda la muestra menos un elemento al azar, utilizando ese elemento para probar el sistema. Luego se repite el cálculo descartando otro elemento diferente, y así sucesivamente.

Con este procedimiento se consigue realizar una prueba que cumple los principios necesarios para probar la propiedad de generalización. Luego, una vez demostrada esta propiedad, se construye el sistema utilizando toda la muestra. Este método presenta el problema de la lentitud, y de que no se prueba el sistema final, sino que se prueban una serie de sistemas similares al final. Sin embargo, es una solución válida cuando la muestra es necesariamente reducida.

[22] *cross-validation* en inglés

5.1.2 Criterios para la selección de características

En general se busca el conjunto mínimo de características que permiten determinar de manera unívoca a qué clase pertenecen todos los objetos del universo de trabajo. Una elección mala de las características discriminantes puede hacer que el sistema sea innecesariamente caro y lento, o que sea imposible construir un clasificador para resolver un problema en base a tales características.

Se pueden exigir cinco propiedades que deben poseer las características que se seleccionen: economía, velocidad, *fiabilidad*, capacidad *discriminante* e *independencia* con respecto a otras características.

Economía

El mecanismo preciso para el cálculo o la obtención de las características discriminantes (sensores, etc..) debe tener un coste razonable.

Velocidad

El tiempo de cálculo no debe superar el umbral que lo haga inviable.

Independencia

Las características no deben estar correladas entre ellas. Una característica que depende fuertemente del resto no añade información discriminante y por tanto puede eliminarse sin que esto suponga ninguna pérdida de capacidad discriminante.

La covarianza $c_{X,Y}$ permite medir la independencia lineal entre dos variables aleatorias X e Y. La estimación de la covarianza entre las variables X e Y de las que hay P ocurrencias responde a la siguiente fórmula:

$$c_{X,Y} = \sum_{p=0}^{P} (X_p - m_X)(Y_p - m_Y)$$

siendo m_X y m_Y los vectores medias correspondientes a los valores de las variables X e Y.

La *matriz de covarianzas* de las características permite medir la independencia lineal de cada par de ellas dentro de una clase. Para ello, si se dispone de N características, se construye una matriz de NxN para cada clase, de manera que cada elemento de la matriz se corresponde con la varianza de un par de características para esa clase. Debe notarse que es necesario disponer de un número significativo de muestras para construir cada una de las matrices, pues en otro caso los resultados no tendrían ninguna validez. Así por ejemplo, para la clase a_k se calcula la matriz C_k, usando las P_k muestras de aprendizaje para esa clase. En ella cada coeficiente c_{ij} corresponde a la covarianza entre la característica i y la j para los patrones de la clase k:

$$C_k = \begin{pmatrix} c_{1,1} & c_{1,2} & \dots & c_{1,N} \\ c_{2,1} & c_{2,2} & \dots & c_{2,N} \\ \dots & \dots & \dots & \dots \\ c_{N,1} & c_{N,2} & \dots & c_{N,N} \end{pmatrix} \quad \text{para la muestra de una clase } a_k$$

Que se desarrolla como:

$$C_k = \frac{1}{P_k} \sum_{p=1}^{P_k} \begin{pmatrix} (X_{p1} - m_{k1})(X_{p1} - m_{k1}) & \dots & (X_{p1} - m_{k1})(X_{pN} - m_{kN}) \\ (X_{p2} - m_{k2})(X_{p1} - m_{k1}) & \dots & (X_{p2} - m_{k2})(X_{pN} - m_{kN}) \\ \dots & \dots & \dots \\ (X_{pN} - m_{kN})(X_{p1} - m_{k1}) & \dots & (X_{pN} - m_{kN})(X_{pN} - m_{kN}) \end{pmatrix}$$

$$\forall \text{ clase } a_k \text{ con } k = 1,2 \dots K \tag{5.1}$$

siendo m_k el vector media de las P_k muestras de la clase k

$$m_k = \frac{1}{P_k} \sum X \qquad \forall\, X \text{ que peretenece a la clase } a_k$$

y siendo m_{kn} la característica n del vector m_k.

El parámetro c_{ij} mide la dependencia lineal entre las características i y j. En vez de usarse c_{ij} suele usarse un parámetro r_{ij} que es independiente de la escala y que se conoce como *coeficiente de correlación*. Este parámetro toma valores entre –1 y 1. Se cumple que si las características i y j son linealmente

independientes, entonces el valor de r_{ij} debe estar próximo a cero[23]. Por otro lado, valores cercanos a –1 ó 1 para r_{ij} indican relaciones lineales entre las características *i* y *j*.

$$r_{ij} = \frac{c_{ij}}{\sqrt{c_{ii}} \cdot \sqrt{c_{jj}}} \quad \forall\, i,j = 1,2 \ldots N$$

Cuando una característica está linealmente correlada con alguna otra, y esto ocurre para todas las clases, puede decidirse eliminarla, ya que ello indica que tal característica no aporta información relevante para el reconocimiento.

Fiabilidad

La fiabilidad implica que objetos de la misma clase deben tener vectores de características con valores numéricos similares. Esto se cumple si los vectores de características de una clase tienen poca *dispersión*. La dispersión se puede medir sobre la diagonal de la matriz de covarianzas. Cuanto mayores son los valores de la diagonal, mayor es la dispersión.

Capacidad discriminante

La capacidad discriminante se puede describir como la exigencia de que los vectores de características de clases distintas tienen que tener valores numéricos distintos.

La capacidad discriminante se puede medir usando el *cociente de Fisher*. La siguiente fórmula permite el cálculo de este parámetro para una característica respecto a dos clases *i* y *j*.

$$F_{ij} = \frac{(m_i - m_j)^2}{s_i^2 + s_j^2}, \text{ donde } s_i = \frac{1}{N}\sum_{p=1}^{P_i}(X_p - m_i)^2$$

[23] La independencia lineal entre dos características *i* y *j* no implica la independencia estadística, ya que pueden existir dependencias no lineales. Por el contrario, la dependencia lineal sí implica dependencia estadística.

Se dice que una característica es tanto más discriminante cuanto mayor es su ratio de Fisher. Se puede ver que este parámetro tiene en cuenta la distancia entre las medias y las desviaciones típicas de las clases de manera inversa. Así, cuanto mayor distancia exista entre los centros de masas de las clases mayor es el ratio, y cuanto menor es la desviación también mayor es el ratio.

La expresión para el caso general de K clases y P_k muestras para cada clase resulta en el siguiente vector F:

$$F = \frac{\frac{1}{K}\sum_{j=1}^{K}(m_j - \overline{m})^2}{\frac{1}{K \cdot P}\sum_{k=1}^{K}\sum_{i=1}^{P_k}(X_{ki} - m_k)^2}$$

siendo

$$\overline{m} = \frac{1}{N}\sum_{j=1}^{P} m_j \qquad \text{y} \qquad P = \sum_{k=1}^{K} P_k$$

5.1.3 Procedimiento de selección

A veces, por motivos de eficiencia, puede ser necesario seleccionar las mejores características. Si las características son estadísticamente independientes se pueden tomar las que posean mayor ratio de Fisher en un primer lugar, y luego, si los criterios de economía y velocidad se cumplen, continuar añadiendo el resto de características por orden.

Si no fuesen independientes, clásicamente, se proponen tres procedimientos: fuerza bruta, eliminación e incorporación. Aunque en la literatura se pueden encontrar multitud de aproximaciones al problema (análisis factorial, análisis discriminante, etc.).

Fuerza bruta

Se construyen todos los conjuntos posibles de características y se selecciona al que mejor cumpla los criterios de selección. Este procedimiento garantiza

encontrar el mejor conjunto de características. En algunos casos es inviable debido a la multitud de combinaciones posibles.

Eliminación

En principio se calcula el rendimiento de *N*-1 clasificadores resultantes de quitar una característica distinta cada vez al conjunto de *N* características. Se elige el vector de características de aquel clasificador en el que la reducción del rendimiento sobre el inicial es menor.

Incorporación

Se van añadiendo características siguiendo el criterio de añadir aquélla que provoca un mayor incremento del rendimiento. Éste es el modo más rápido de obtener un buen conjunto, aunque generalmente no se obtiene el óptimo.

5.2. Tipología de los algoritmos de clasificación de patrones

Los clasificadores se pueden ordenar atendiendo a diferentes criterios (la forma de construirse, el tipo de muestra, la información disponible, etc.). En los siguientes puntos se presentarán los principales criterios.

5.2.1 Clasificadores a priori y a posteriori

Los algoritmos de reconocimiento se pueden dividir en dos grupos atendiendo a la forma en que se construyen los clasificadores: a *priori* y a *posteriori* o con *aprendizaje*.

Los *clasificadores apriorísticos* construyen el clasificador en un solo paso, utilizando la muestra de aprendizaje para el cálculo de las funciones discriminantes.

Los *clasificadores a posteriori* se construyen siguiendo un procedimiento iterativo, en el cual el clasificador aprende a reconocer de una manera progresiva los patrones de la muestra de aprendizaje.

5.2.2 Clasificadores deterministas y no deterministas

Atendiendo a la forma en que se distribuyen los patrones de la muestra se puede hablar de que se cumple o no la hipótesis determinista: cada clase se puede representar por un único vector que se llama prototipo representante de la clase. Según esta hipótesis se puede hablar de dos tipos de clasificadores: *clasificadores deterministas* y *clasificadores no deterministas*.

Dependiendo de las características seleccionadas puede ser necesario el uso de uno u otro tipo. Cuando las características elegidas hacen que los patrones de clases diferentes se sitúen en regiones disjuntas, los clasificadores deterministas darán buenos resultados. Si las regiones no son disjuntas ofrecerán mejores resultados los clasificadores no deterministas.

5.2.3 Clasificadores supervisados y no supervisados

Atendiendo a la información que se proporciona en el proceso de construcción del clasificador se puede hablar de dos tipos de clasificadores: con *maestro* o *supervisados*, sin maestros o *no supervisados*.

En los supervisados, la muestra la divide el maestro en las diferentes clases ya conocidas en las que se desea clasificar. A grandes rasgos las etapas en la construcción de un clasificador con maestro son: determinación de las clases, elección y test de las características discriminantes, selección de la muestra, cálculo de funciones discriminantes y test del clasificador.

En los no supervisados este proceso se realiza de manera automática, sin la necesidad de ningún supervisor externo. Para ello se emplean técnicas de agrupamiento, gracias a las cuales el sistema selecciona y aprende los patrones que poseen características similares, determinándose automáticamente las clases.

En el siguiente punto se tratan tres clasificadores básicos que cubren diferentes tipos de clasificadores supervisados. En el se hablará del clasificador a priori por distancia euclídea, del clasificador a priori de Bayes y del clasificador euclídeo con aprendizaje. Seguidamente, en el apartado 5.4, se plantean diferentes técnicas no supervisadas de agrupamiento de clases.

5.3. Clasificadores basados en la distancia

Existen multitud de clasificadores diferentes. Algunos, como los que se van a estudiar en este capítulo, se basan en el concepto de distancia entre los vectores de características. Sin embargo, hay que decir que existen muchos otros enfoques. Así, por ejemplo, están aquéllos que se basan en medidas de probabilidad de pertenencia a una u otra clase (como los clasificadores basados en Modelos Ocultos de Markov), o los que usan lógica borrosa para la clasificación en base a reglas, o a los que utilizan redes de neuronas artificiales (ver el apéndice A).

A continuación, se exponen diferentes clasificadores basados en distancias, que permiten por su simplicidad y variedad una aproximación didáctica y adecuada al problema de la clasificación.

5.3.1 Clasificador de distancia euclídea determinista a priori

Es éste un clasificador determinístico, supervisado y a priori. Se basa en el cálculo de un *prototipo* o *centroide* para cada una de las K clases en las que se divide el universo de trabajo. Este prototipo puede verse como un representante "ejemplar" de cómo debería de ser un vector de características de esa clase. Así, ante un patrón desconocido se calcula la distancia euclídea del patrón que se desea clasificar a cada uno de los K prototipos. Un patrón desconocido X se clasificará como correspondiente a la clase cuyo prototipo esté a menor distancia según la distancia euclídea:

$$d_E(X, Z_k) = \sqrt{X^T \cdot X - 2 \cdot X^T \cdot Z_k + Z_k^T \cdot Z_k} \qquad (5.1)$$

Así, el *clasificador euclídeo* divide el espacio de características en regiones mediante hiperplanos equidistantes de los centroides. La figura adjunta presenta el caso de 3 clases y 2 características. Cada línea punteada separa los puntos del espacio más cercano a cada uno de los centroides (representados por puntos negros).

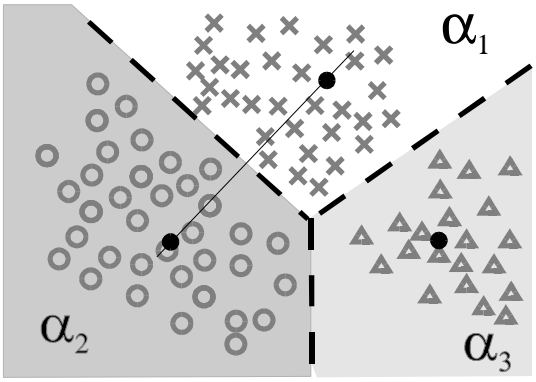


Figura 85.- Ejemplo de separación lineal entre clases.

Así, para el caso de K clases: a_1, a_2, ... a_K, se necesitan K prototipos: Z_1, Z_2 ... Z_K. Al clasificar un patrón se sigue el esquema de la Figura 86. Según este esquema, para clasificar el patrón X se calcula la distancia del vector de características X a los vectores de características de cada uno de los K prototipos (Z_1, Z_1 ... Z_K), clasificando X como perteneciente a la clase cuyo prototipo esté más próximo.

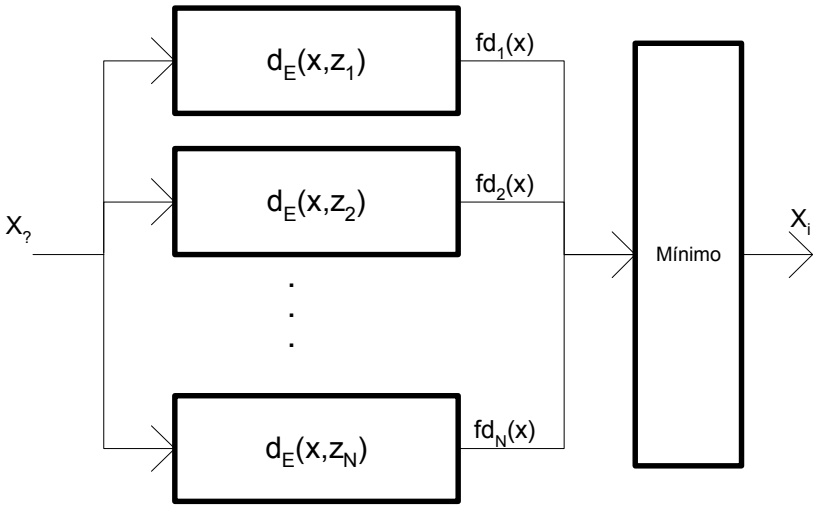


Figura 86.- Esquema del clasificador euclídeo.

La función discriminante (5.1) puede simplificarse eliminando la raíz cuadrada, ya que es una transformación que mantiene la relación de distancia. También se puede eliminar el término $X^T X$ pues es igual para todas las clases. Finalmente suele cambiarse el signo y dividirse por 2 obteniendo la expresión (5.2). Debido al cambio de signo será ahora la función discriminante con valor máximo la que marcará la clase a la que pertenece el patrón *X*.

$$fd_k(X) = X^T \cdot Z_k - \frac{1}{2} Z_k^T \cdot Z_k \qquad \forall\, k = 1 \ldots K \tag{5.2}$$

El proceso de cálculo de los prototipos es un proceso heurístico. El método que suele usarse para construir el prototipo de una clase consiste en el cálculo de la media ponderada de un conjunto de *P* elementos de esa clase. Esto hace que para el cálculo de prototipos se necesite un conjunto de muestra que incluya varios individuos para cada una de las *N* clases. El tamaño de este conjunto debe ser tal que sea representativo de la clase de elementos a la que corresponda.

$$Z_k = \frac{1}{P_k} \sum_{p=1}^{P_k} X_p \qquad \forall\, X_p \text{ patrón de la clase } a_k$$

Ejemplo 15.-

Supóngase que se tienen dos clases a_1 y a_2 con centroides:

$$z_1 = \begin{pmatrix} 1 \\ 3 \\ 5 \\ 1 \end{pmatrix} \qquad z_2 = \begin{pmatrix} 7 \\ 8 \\ 1 \\ 7 \end{pmatrix}$$

Las funciones discriminantes asociadas a cada clase son:

$$fd_1 = \vec{X}^T \begin{pmatrix} 1 \\ 3 \\ 5 \\ 1 \end{pmatrix} - \frac{1}{2} \begin{pmatrix} 1 \\ 3 \\ 5 \\ 1 \end{pmatrix}^T \begin{pmatrix} 1 \\ 3 \\ 5 \\ 1 \end{pmatrix} \qquad \text{y} \qquad fd_2 = \vec{X}^T \begin{pmatrix} 7 \\ 8 \\ 1 \\ 7 \end{pmatrix} - \frac{1}{2} \begin{pmatrix} 7 \\ 8 \\ 1 \\ 7 \end{pmatrix}^T \begin{pmatrix} 7 \\ 8 \\ 1 \\ 7 \end{pmatrix}$$

Así, el vector X=(3,1,3,1) , pertenece a la clase a_1 pues.

$$fd_1 = \begin{pmatrix}3\\1\\3\\1\end{pmatrix}^T \begin{pmatrix}1\\3\\5\\1\end{pmatrix} - \frac{1}{2}\begin{pmatrix}1\\3\\5\\1\end{pmatrix}^T \begin{pmatrix}1\\3\\5\\1\end{pmatrix} = 3+3+15+1-\frac{1}{2}(1+9+25+1) = 4$$

$$fd_2 = \begin{pmatrix}3\\1\\3\\1\end{pmatrix}^T \begin{pmatrix}7\\8\\1\\7\end{pmatrix} - \frac{1}{2}\begin{pmatrix}7\\8\\1\\7\end{pmatrix}^T \begin{pmatrix}7\\8\\1\\7\end{pmatrix} = 21+8+3+7-\frac{1}{2}(49+64+1+49) = -42.5$$

de manera que $fd_1(X) > fd_2(X) \Rightarrow X \in \alpha_1$

5.3.2 Clasificador estadístico a priori

En las situaciones en que los vectores de alguna clase presenten una dispersión significativa respecto a la media, o en aquéllas en las que no existe posible separación lineal entre las clases, puede ofrecer mejores resultados la sustitución de la distancia euclídea por la *distancia de Mahalanobis*. Esta medida, que tiene en cuenta la desviación típica de los vectores de características de los patrones de la muestra, puede proporcionar regiones de separación entre clases que sigan curvas cónicas. Por ello, este clasificador ofrece más garantías al tratar de separar patrones para los que no se encuentra una separación lineal. Además se verá que el c*lasificador estadístico* proporciona probabilidades de pertenencia a las clases, por lo que se debe incluir en el grupo de los clasificadores no deterministas.

La distancia de Mahalanobis de un punto X a una clase a_k viene dada por la expresión (5.3), donde m_k corresponde a la media de la clase k y C_k a la matriz de covarianzas. En el siguiente punto se explicará cómo aparece esta distancia de manera natural al enfocar el problema de la clasificación desde un punto de vista probabilista.

$$d_M(X, a_k) = (X - m_k)^T \cdot C_k^{-1} \cdot (X - m_k) \tag{5.3}$$

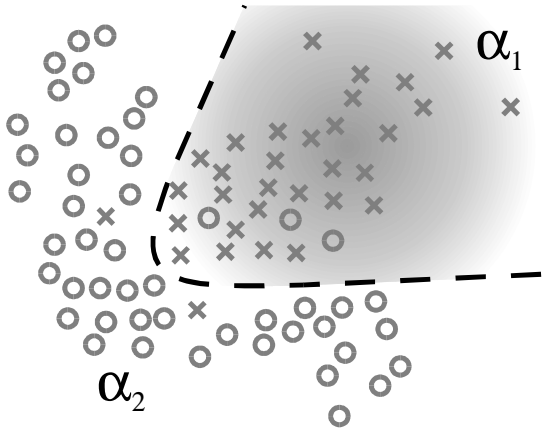


Figura 87.- En un clasificador estadístico la región de separación no es determinista. Además crea separaciones no lineales de dos clases (curvas cónicas como las parábolas, elipses, hipérbolas y, por su puesto, rectas).

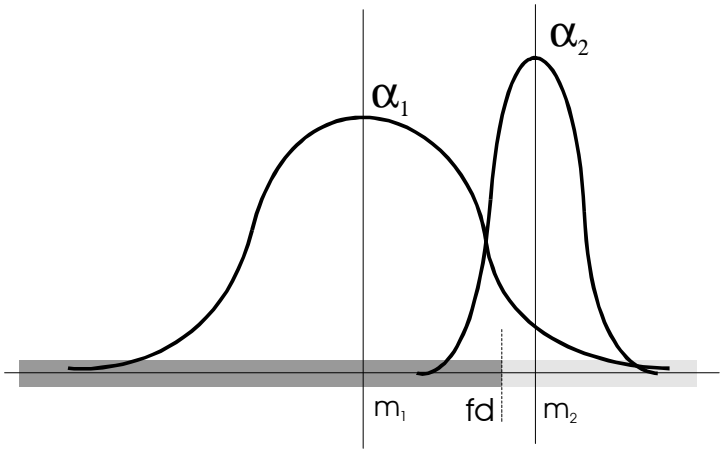


Figura 88.- La función discriminante depende de la desviación típica de las clases.

Clasificación probabilista

Cuando existe diferente dispersión de los valores de una característica en dos o más clases una medida de la distancia que tenga en cuenta la desviación típica ofrecerá mejores resultados. La Figura 88 ejemplifica la disposición de los patrones de dos clases respecto de una característica particular. En la figura se

aprecia que la función discriminante no equidista de los centroides del α_1 y de α_2 sino que está más cerca del centroide de α_2 porque su desviación típica es menor.

El teorema de Bayes enuncia que:

$$P(a_i / X) = \frac{P(X / a_i) \cdot P(a_i)}{P(X)}$$

donde,

- $P(a_i)$ es la probabilidad de que un patrón al azar sea de la clase a_i.
- $P(X)$ es la probabilidad de que se presente exactamente un patrón con el vector de características X. Cumpliéndose que:

$$P(X) = \sum_{k=1}^{K} P(X / a_k) P(a_k)$$

- $P(X/a_i)$ es la probabilidad de que sabiendo de que el patrón a clasificar pertenece a la clase a_i sea precisamente el que tiene el vector de características X. En otras palabras es la función de densidad de la clase a_i y se denomina *probabilidad a priori*.
- $P(a_i/X)$ es la probabilidad de que un X conocido pertenezca a la clase a_i. Se denomina *probabilidad a posteriori*.

Aunque el clasificador estadístico es no determinista, cuando se precisa una respuesta concreta suele transformarse en determinista añadiendo la siguiente regla:

$$X \in a_i \Leftrightarrow P(a_i /X) > P(a_j /X) \quad \forall\, i \neq j, j = 1,2 \ldots N$$

Como $P(X)$ es un término constante para todos los cálculos de $P(a_i /X)$, el cálculo puede simplificarse eliminando ese factor. Con esto la función discriminante para la clase a_i es:

$$fd_i(X) = P(X/a_i) \times P(a_i) \qquad \forall\, i = 1,2 \ldots N$$

Y se dice que:

$$X \in a_i \Leftrightarrow fd_i(X) > fd_j(X) \quad \forall\, i \neq j, j = 1,2 \ldots N$$

Diseño de un clasificador estadístico a priori

El diseño de un clasificador exige conocer la distribución de probabilidad $P(X/a_i)$. Para el caso más simple la distribución de probabilidad $P(X/a_i)$ sigue una distribución normal o gausiana unidimensional. Es por esto que suele hacerse la hipótesis de normalidad. Caso de no cumplirse debe proponerse una distribución de probabilidad adecuada. El siguiente desarrollo supone la hipótesis de normalidad.

La siguiente fórmula presenta la función de probabilidad para un vector una sola característica con una distribución de probabilidad normal.

$$P(X/a_i) = \frac{1}{\sqrt{2p}\cdot s_i} e^{-\frac{1}{2}\frac{(X-m_i)^2}{s_i^2}} \qquad \forall\, \text{i} = 1,2 \ldots \text{N}$$

siendo m_i la media y s_i la desviación típica de la distribución para la clase a_i.

Si la probabilidad de obtener objetos de diferentes clases es la misma los cálculos se simplifican. Ya que si $P(a_i) = P(a_k)$ para cualquier clase i y k es termino no influye en el cálculo. Así, para el caso de dos clases se obtiene:

$$X \in a_1 \Leftrightarrow P(a_1/X) > P(a_2/X) \Leftrightarrow P(X/a_1) > P(X/a_2)$$

Esta expresión puede simplificarse tomando logaritmos:

$$\frac{(X-m_1)^2}{s_1^2} < \frac{(X-m_2)^2}{s_2^2} + 2\ln\left(\frac{s_2}{s_1}\right)$$

Para el caso n-dimensional la función de probabilidad normal es:

$$P(X/a_k) = \frac{1}{(2p)^{\frac{n}{2}} \cdot |C_k|^{\frac{1}{2}}} e^{-\frac{1}{2}(\vec{X}-\vec{m}_k)^T \cdot C_k^{-1} \cdot (\vec{X}-\vec{m}_k)} \qquad \forall\, k = 1,2 \ldots N$$

siendo C_k la matriz de covarianzas (5.1).

Teniendo en cuenta los resultados precedentes, la función general discriminante para una clase, cuando las probabilidades de ocurrencia de las diferentes clases son iguales, y tras hacer logaritmos para eliminar el termino exponencial, resulta:

$$fd_i(X) = -\frac{1}{2}\vec{X}^T C_i^{-1}\vec{X} + \vec{X}^T C_i^{-1}\vec{m}_i - \frac{1}{2}\vec{m}_i^T C_i^{-1}\vec{m}_i - \frac{1}{2}\ln|C_i| \quad \forall\, i = 1,2 \ldots N$$

Puede comprobarse que en el caso de que las matrices de covarianza fuesen iguales la función discriminante se simplifica a:

$$fd_i(X) = \vec{X}^T C_i^{-1}\vec{m}_i - \frac{1}{2}\vec{m}_i^T C_i^{-1}\vec{m}_i \quad \forall\, i = 1,2 \ldots N$$

Si además la matriz de covarianza es diagonal, con covarianzas a cero, y con todas las desviaciones típicas iguales se obtiene una formula idéntica a la del clasificador euclídeo:

$$fd_i(X) = \vec{X}^T \vec{m}_i - \frac{1}{2}\vec{m}_i^T m_i \quad \forall\, i = 1,2 \ldots N$$

Hay que señalar que en la práctica los valores que se obtienen para las desviaciones nunca son exactamente iguales, ni las covarianzas son exactamente cero, pero estas reglas se aplican igualmente si se aproximan suficientemente a tales valores.

Ejemplo 16.-

Se dispone de la siguiente muestra, correspondiente a objetos de dos clases a_1 y a_2 equiprobables. Se desea construir un clasificador de Bayes que clasifique correctamente los patrones de la muestra sabiendo que los patrones tienen características que siguen distribuciones normales.

$$a_1 : \left\{\begin{pmatrix}1\\2\end{pmatrix}, \begin{pmatrix}2\\2\end{pmatrix}, \begin{pmatrix}3\\1\end{pmatrix}, \begin{pmatrix}2\\3\end{pmatrix}, \begin{pmatrix}3\\2\end{pmatrix}\right\} \quad a_2 : \left\{\begin{pmatrix}8\\10\end{pmatrix}, \begin{pmatrix}9\\8\end{pmatrix}, \begin{pmatrix}9\\9\end{pmatrix}, \begin{pmatrix}8\\9\end{pmatrix}, \begin{pmatrix}7\\9\end{pmatrix}\right\}$$

Sobre la muestra anterior se calculan las medias y las matrices de covarianza que resultan ser:

$$\vec{m}_1 = \begin{pmatrix} 2.2 \\ 2 \end{pmatrix} \quad \vec{m}_2 = \begin{pmatrix} 8.2 \\ 9 \end{pmatrix} \quad C_1 = \begin{pmatrix} 0.56 & -0.2 \\ -0.2 & 0.4 \end{pmatrix} \quad C_2 = \begin{pmatrix} 0.56 & -0.2 \\ -0.2 & 0.4 \end{pmatrix}$$

Como se observa que $C_1 = C_2$ y además se ha dicho en el enunciado que los patrones de ambas clases son equiprobables y sus distribuciones normales se puede usar la función discriminante:

$$fd_i(X) = X^T C^{-1} m_i - \frac{1}{2} m_i^T C^{-1} m_i$$

obteniendo:

$$C^{-1} = \begin{pmatrix} \frac{50}{23} & \frac{25}{23} \\ \frac{25}{23} & \frac{70}{23} \end{pmatrix}$$

y por tanto:

$$fd_1(X) = \frac{160}{23} X_1 + \frac{195}{23} X_2 - \frac{378}{23}$$

$$fd_2(X) = \frac{435}{23} X_1 + \frac{835}{23} X_2 - \frac{6361}{23}$$

5.3.3 Clasificador de distancia con aprendizaje supervisado

El problema de la clasificación de un vector X en una de K clases puede plantearse como un problema de optimización. Más formalmente, para un conjunto de K clases $\{a_1, a_2, \ldots a_K\}$ la solución al problema de clasificar un vector X desconocido consiste en asociarlo a aquella clase cuya función discriminante $fd_i(X)$ dé un resultado máximo, es decir:

$$x \in a_k \iff fd_k(x) > fd_j(x) \qquad \forall\, j = 1, 2, \ldots, K \;/\; j \neq k$$

Estas funciones discriminantes, para el caso de la distancia, son funciones lineales de la forma:

$$fd_k(X) = W_k^T \times X \qquad \forall\, k = 1, 2, \ldots, K$$

donde X es el vector de características al que se le ha añadido un valor de 1 para contemplar en la notación matricial el termino independiente de las ecuaciones lineales. Por tanto, el objetivo es encontrar los valores W_k que hagan de fd_k una función discriminante.

Clásicamente, para resolver estos problemas suelen usarse algoritmos de *descenso de gradiente*. Estos algoritmos parten de ecuaciones con valores al azar y mejoran sucesivamente los parámetros de estas ecuaciones hasta alcanzar una solución al problema (generalmente no óptima). En el argot del reconocimiento de patrones este procedimiento de mejora sucesiva se denomina "*aprendizaje*".

La diferencia con los reconocedores apriorísticos (sin aprendizaje) consiste en que en aquéllos se calculaba a priori las funciones discriminantes, mientras que en éstos, las funciones discriminantes se calculan mediante un procedimiento iterativo. Los procesos de aprendizaje ofrecen ventajas reales cuando las ecuaciones son no lineales y calcular la solución analítica no es posible (como en el caso de las redes de neuronas). Aquí se expone usando el clasificador euclídeo por motivos didácticos.

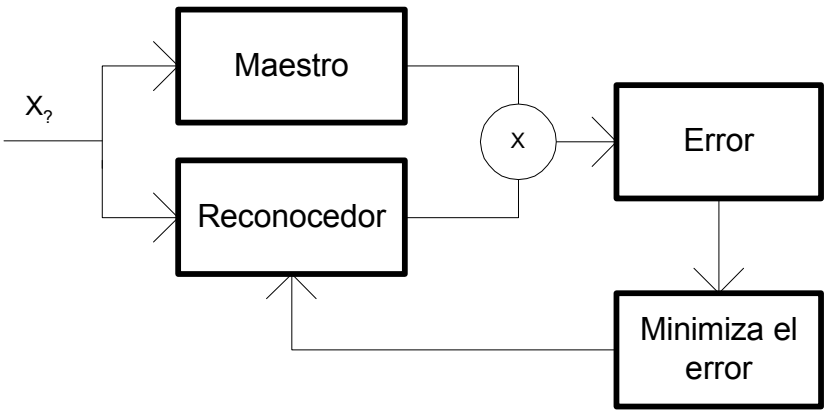


Figura 89.- Esquema de reconocimiento con aprendizaje. El maestro permite calcular el error y así decidir cómo modificar los parámetros del clasificador para minimizarlo.

Algoritmo de aprendizaje

Los algoritmos de aprendizaje parten de unas funciones discriminantes al azar y las van mejorando sucesivamente hasta que obtienen funciones discriminantes que clasifican correctamente la mayor parte de los patrones del universo de trabajo.

En un instante cualquiera se tienen unas funciones discriminantes que cometen un cierto error al clasificar los patrones de aprendizaje. El algoritmo de aprendizaje se encarga de cambiar los valores de los parámetros de las funciones discriminantes para que se reduzca el error que se comete al reconocer. Este proceso se repite iterativamente. Así, partiendo de unas funciones discriminantes aleatorias se va disminuyendo el error que se comete al clasificar los patrones de manera progresiva.

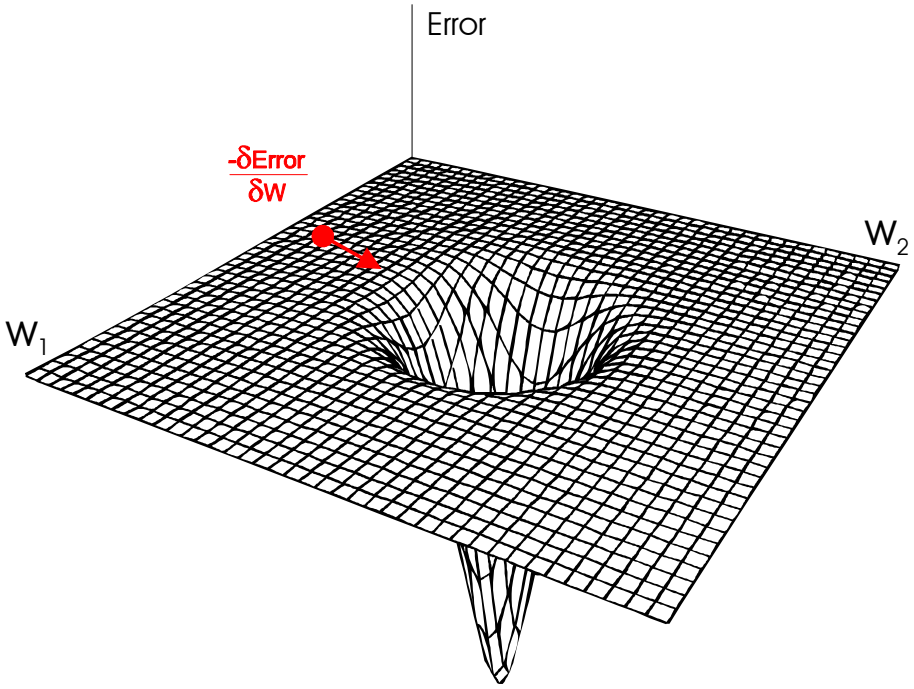


Figura 90.- El error debe verse como una superficie función de los parámetros *W* del clasificador. Para unos *W* dados se obtiene un error determinado. La derivada marca la dirección con la que disminuye el error.

El error cometido depende de lo que se quiere obtener (el patrón clasificado de acuerdo a lo que fija el maestro) y lo que se obtiene con las funciones discriminantes en su estado actual.

Error = diferencia entre lo que se desea obtener y lo que se obtiene

El error se puede visualizar como una hipersuperficie que es función de los valores de W, que son los parámetros del clasificador. Por tanto, el error sólo depende de los valores de los vectores W_i pues son los que determinan las funciones discriminantes. En la Figura 90 se presenta un ejemplo en el que solo hay dos características y por tanto la superficie tiene dos dimensiones.

La derivada de la función *Error* respecto de W marca la dirección de máxima pendiente dentro de esa superficie. Por ello, la dirección en la que se reduce el error viene determinada por la dirección de la derivada. Así, si se deriva la función del error respecto de W se obtiene la dirección en que hay que variar W para que disminuya el error. El proceso de cambio de los W para reducir el error, es decir el aprendizaje, consiste en saltar un factor m de veces en la dirección de la derivada sobre la superficie del error. Es decir:

$$W(t+1) = W(t) - m\frac{\partial Error}{\partial W} \tag{5.4}$$

El factor m determina la velocidad del aprendizaje. Es un valor que se debe determinar de manera empírica (normalmente por ensayo y error) para cada problema, ya que depende de la superficie del error que se esté considerando, y por tanto de la muestra del problema en cuestión. La elección de un valor pequeño para m puede hacer que el aprendizaje sea lento, un valor grande puede hacer que se salten soluciones (al no entrar dentro de los "agujeros" de la hipersuperficie). Desgraciadamente los conceptos de "grande" y "pequeño" dependen de cada problema particular, pues al cambiar los patrones de entrenamiento cambia la superficie de error.

Para el caso que nos ocupa se toma la siguiente función para el error:

$$Error = \frac{1}{4}(fdm - W_i^T X)^2$$

siendo:

$$fdm = \begin{cases} +1 & \text{si} \quad X \in a_i \\ -1 & \text{si} \quad X \notin a_i \end{cases}$$

De esta forma y llamando *e* a:

$$e = (fdm - W_i^T X)$$

se obtiene que:

$$\nabla Error = \frac{\partial Error}{\partial W} = \frac{1}{4} 2(fdm - W_i^T X)(-X) \Rightarrow \nabla Error = -\frac{1}{2} eX$$

y por tanto:

$$W(t+1) = W(t) + m\frac{1}{2} eX$$

Esta expresión muestra como deben cambiarse los pesos de todas las funciones discriminantes en cada iteración para que el error disminuya progresivamente. El algoritmo se repite hasta que el error cae por debajo de un mínimo admisible, o hasta que los vectores *W* converjan.

El algoritmo de aprendizaje queda detallado en el siguiente pseudocódigo.

- Algoritmo de aprendizaje para el clasificador euclídeo -

Paso1.- Se dispone un conjunto de *P* muestras de aprendizaje o conjunto de entrenamiento.

$$CE = \{X_0, X_1, \ldots X_{P-1}\}$$

Se inician al azar los valores de W_i. Se inicia $t = 0$ y $p = 0$. Se fija un nivel de error máximo aceptable *E*.

Paso 2.- Se presenta la muestra de entrenamiento X_p (que el maestro sabe que pertenece a a_k) y se calculan las *K* funciones discriminantes.

$$fd_1(X_p),\ fd_2(X_p),\ \ldots fd_K(X_p)$$

Paso 3.- Para cada *fd* calcular el error que se comete. Si el error es menor al *E* fijado PARAR, en otro caso actualizar *W* según:

$$W(t+1) = W(t) + 0'5\ m\ .\ \text{error}\ .\ X$$

Ir al paso 2.

Ejemplo 17.-

Determinar los vectores W_1, W_2 y W_3 de las funciones discriminantes fd_1, fd_2 y fd_3 para el problema de clasificación de las tres clases dadas por los siguientes tres patrones.

$$x_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \in a_1, x_2 = \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} \in a_2, x_3 = \begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix} \in a_3$$

Utilizando el algoritmo de aprendizaje descrito, con μ=2 y partiendo en t=0 de:

$$W_1(0) = W_2(0) = W_3(0) = \begin{pmatrix} 0 & 0 & 0 \end{pmatrix}$$

Se obtiene en $t = 1$:

$X(1) = x_1 \ \hat{I}\ a_1$

$fd_1(X(1)) = W_1(1)^T \times X(1) = 0$

$fd_2(X(1)) = W_2(1)^T \times X(1) = 0$

$fd_3(X(1)) = W_3(1)^T \times X(1) = 0$

$error_0(1) = 1$

$error_1(1) = -1$

$error_2(1) = -1$

$W_1(2) = W_1(1) + e \times m \times X(1) = (1, 0, 0)$

$W_2(2) = W_2(1) - e \times m \times X(1) = (-1, 0, 0)$

$W_3(2) = W_3(1) - e \times m \times X(1) = (-1, 0, 0)$

Se obtiene en $t = 2$:

$X(2) = x_2 \ \hat{I}\ a_2$

$fd_1(X(2)) = W_1(2)^T \times X(2) = 0$

$fd_2(X(2)) = W_2(2)^T \times X(2) = 0$

$fd_3(X(2)) = W_3(2)^T \times X(2) = 0$

$error_1(2) = -1$

$error_2(2) = 1$

$error_3(2) = -1$

$W_1(3) = W_1(2) - e \times m \times X(2) = (1, -1, -1)$

$W_2(3) = W_2(2) + e \times m \times X(2) = (-1, 1, 1)$

$W_3(3) = W_3(2) - e \times m \times X(2) = (-1, -1, -1)$

Se obtiene en $t = 3$:

$X(3) = x_3 \ \hat{I}\ a_3$

$fd_1(X(3)) = W_1(3)^T \times X(3) = 1$

$fd_2(X(3)) = W_2(3)^T \times X(3) = -1$

$fd_3(X(3)) = W_3(3)^T \times X(3) = 1$

$error_1(3) = -2$

$error_2(3) = 0$

$error_3(3) = 0$

$W_1(4) = W_1(3) - e \cdot X(3) = (1, -1, 1)$

$W_2(4) = W_2(3) = (-1, 1, 1)$

$W_3(4) = W_3(3) = (-1, -1, -1)$

Se obtiene en $t = 4$:

$X(4) = x_1 \in a_1$

$fd_1(X(4)) = W_1(4)^T \cdot X(4) = 1$

$fd_2(X(4)) = W_2(4)^T \cdot X(4) = -1$

$fd_3(X(4)) = W_3(4)^T \cdot X(4) = -1$

$error_1(4) = 0$

$error_2(4) = 0$

$error_3(4) = 0$

$W_1(5) = W_1(4) = (1, -1, 1)$

$W_2(5) = W_2(4) = (-1, 1, 1)$

$W_3(5) = W_3(4) = (-1, -1, -1)$

Se obtiene en $t = 5$:

$X(5) = x_2 \notin a_2$

$fd_1(X(5)) = W_1(5)^T \cdot X(5) = 0$

$fd_2(X(5)) = W_2(5)^T \cdot X(5) = 2$

$fd_3(X(5)) = W_3(5)^T \cdot X(5) = -2$

$error_1(5) = -1$

$error_2(5) = -1$

$error_3(5) = 1$

$W_1(6) = W_1(5) - e \cdot X(5) = (1, -2, 0)$

$W_2(6) = W_2(5) - e \cdot X(5) = (-1, 0, 0)$

$W_3(6) = W_3(5) + e \cdot X(5) = (-1, 0, 0)$

Se obtiene en $t = 6$:

$X(6) = x_3 \in a_3$

$fd_1(X(6)) = W_1(6)^T \cdot X(6) = 0$

$fd_2(X(6)) = W_2(6)^T \cdot X(6) = 0$

$fd_3(X(6)) = W_3(6)^T \cdot X(6) = 0$

$error_1(6) = -1$

$error_2(6) = -1$

$error_3(6) = 1$

$W_1(7) = W_1(6) - e \cdot X(6) = (1, -2, 1)$

$W_2(7) = W_2(6) - e \cdot X(6) = (-1, 0, 1)$

$W_3(7) = W_3(6) + e \cdot X(6) = (-1, 0, -1)$

Se obtiene en $t = 7$:

$X(7) = x_1 \in a_1$

$fd_1(X(7)) = W_1(7)^T \cdot X(7) = 1$

$fd_2(X(7)) = W_2(7)^T \cdot X(7) = -1$

$fd_3(X(7)) = W_3(7)^T \cdot X(7) = -1$

$error_1(7) = 0$

$error_2(7) = 0$

$error_3(7) = 0$

$W_1(8) = W_1(7) = (1, -2, 1)$

$W_2(8) = W_2(7) = (-1, 0, 1)$

$W_3(8) = W_3(7) = (-1, 0, -1)$

Se obtiene en $t = 8$:

$X(8) = x_2 \in a_2$

$fd_1(X(8)) = W_1(8)^T \cdot X(8) = -1$

$fd_2(X(8)) = W_2(8)^T \cdot X(8) = 1$

$fd_3(X(8)) = W_3(8)^T \cdot X(8) = -1$

$error_1(8) = 0$

$error_2(8) = 0$

$error_3(8) = 0$

$W_1(9) = W_1(8) = (1, -2, 1)$

$W_2(9) = W_2(8) = (-1, 0, 1)$

$W_3(9) = W_3(8) = (-1, 0, -1)$

Se obtiene en $t = 9$:

$X(9) = x_3 \in a_3$

$fd_1(X(9)) = W_1(9)^T \cdot X(9) = -1$

$fd_2(X(9)) = W_2(9)^T \cdot X(9) = -1$

$fd_3(X(9)) = W_3(9)^T \cdot X(9) = 1$

$error_1(9) = 0$

$error_2(9) = 0$

$error_3(9) = 0$

$W_1(9) = W_1(8) = (1, -2, 1)$

$W_2(9) = W_2(8) = (-1, 0, 1)$

$W_3(9) = W_3(8) = (-1, 0, -1)$

Como se ha completado una pasada del clasificador con todos los patrones de la muestra y no han cambiado los coeficientes *W* se termina el proceso de entrenamiento. Las funciones discriminantes obtenidas son:

$$fd_1(X) = \begin{pmatrix} 1 & -2 & 1 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ 1 \end{pmatrix} = x_1 - 2x_2 + 1$$

$$fd_2(X) = \begin{pmatrix} -1 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ 1 \end{pmatrix} = -x_1 + 1$$

$$fd_3(X) = \begin{pmatrix} -1 & 0 & -1 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ 1 \end{pmatrix} = -x_1 - 1$$

Nótese lo insuficiente de la muestra de entrenamiento del ejemplo, que sólo tiene 3 patrones habiendo 3 clases. En este ejemplo se ha usado el mínimo número de patrones preciso para entrenar. Normalmente se dispone de varios patrones de muestra por cada clase durante la fase de entrenamiento. Obviamente esta simplificación se debe a motivos didácticos, ya que un conjunto mayor no puede presentarse en poco espacio.

5.3.4 Clasificador de *k*-vecinos más cercanos

En aquellos casos en los que la región de clasificación no es lineal ni sigue una curva cónica puede usarse el *clasificador de los k-vecinos* más cercanos. Intuitivamente este clasificador supervisado apriorístico es uno euclídeo con múltiples centroides por cada clase.

La principal ventaja de este clasificador reside en su potencia. Su principal desventaja estriba en la dificultad para la determinación del número de centroides por clase y su disposición, tareas que en general son eminentemente empíricas. Además el tiempo de cálculo puede ser significativamente mayor si el número de centroides llega a ser muy elevado.

5.4. Algoritmos de agrupación de clases

Hasta ahora se ha tratado algunos métodos de clasificación supervisados. En ocasiones no existe la figura del maestro que determina los patrones que pertenecen a una clase o a otra. En estos casos los algoritmos de *agrupamiento* o *clustering* permiten realizar esta tarea de manera automática. Estos algoritmos también se conocen como algoritmos de clasificación autoorganizados.

Los algoritmos de agrupación de clases se suelen utilizar cuando no existe conocimiento a priori de las clases en que se pueden distribuir los objetos, cuando las clases no son interpretables por un humano, o cuando el número de clases es muy elevado para un procesado no automático. En los siguientes puntos se exponen el algoritmo de *distancias encadenadas*, el algoritmo *MaxMin* y el algoritmo de las *k-medias*. No queremos terminar este punto introductorio sin citar otros algoritmos clásicos en la problemática de la agrupación de clases como los dendrogramas o los algoritmos de clustering jerárquico, aunque no los abordaremos por su ineficacia en problemas en los que la muestra tiene un tamaño grande.

5.4.1 Algoritmo de distancias encadenadas

El *algoritmo de las distancias encadenadas* es un algoritmo que no precisa información sobre el número de clases existente. Tan sólo necesita el ajuste de un umbral que fijará el nivel de histéresis en la determinación de las clases, es decir aquel valor que al ser superado produce un cambio de clase.

Este algoritmo parte de los vectores de características de la muestra de aprendizaje X_1, X_2,...X_P y toma uno de ellos X_i al azar. Seguidamente ordena los vectores según la sucesión:

$$X_i(0), X_i(1), X_i(2)...X_i(P\text{-}1)$$

donde esta sucesión es tal que el siguiente vector de la cadena es el más próximo al anterior según la distancia euclídea. Es decir $X_i(1)$ es el más próximo *a* $X_i(0)$, $X_i(2)$ es el más próximo a $X_i(1)$, etc. Siendo $X_i(0) = X_i$.

Finalmente se elige un valor umbral, que está relacionado con la distancia máxima que puede haber entre elementos de una clase. El primer elemento pertenece a la clase 0. Si la distancia entre dos elementos consecutivos de la

sucesión es superior al umbral, en ese punto de la sucesión, comienza una nueva clase, en otro caso, el elemento analizado pertenece a la misma clase que el elemento anterior.

- Algoritmo Chain Map -

Paso 1.- Se saca un elemento X al azar del conjunto de muestras de aprendizaje M y se le llama V. Se mete V en $C[0]$. Se inicia $a = 0$.

Paso 2.- Se calcula la distancia de V al resto de elementos de M y se toma el elemento X con menor distancia a V.

Paso 3.- Si la distancia de X a V es mayor al umbral se hace $a = a + 1$.

Paso 4.- Se mete en la clase $C[a]$, se hace $V = X$ y se elimina X de M.

Paso 5.- Si M está vacío se finaliza, en otro caso ir al paso 2.

Este algoritmo tiene la ventaja de no precisar información sobre número de clases, y de que se realiza en un solo paso, por lo puede ser muy rápido. Sus principales inconvenientes se encuentran en la dificultad de fijar el valor del umbral, y en que la solución que se obtiene puede depender del punto de inicio del algoritmo.

5.4.2 Algoritmo MaxMin

El *algoritmo MaxMin* introduce paulatinamente representantes, determinando la distancia a la que se encuentran los patrones de los mismos. Si en algún caso esta distancia supera un umbral se crea una nueva clase. El proceso se repite hasta que no se producen cambios.

Es un algoritmo que sólo precisa la determinación de un umbral para la autoorganización de las clases, además este valor determina el radio de las clases entorno al representante de cada una. Este algoritmo tiene dos desventajas principales: la de ser algo frágil respecto a la elección del punto inicial, y la dificultad de fijar el valor del umbral a priori.

- Algoritmo Max Min -

Paso 1.- Se fija una valor umbral *h*. Se toma al azar un elemento de los *P* disponibles y se toma como representante de la clase a_1.

Paso 2.- Se calculan las distancias euclídeas del representante de a_1 a los *P*-1 vectores no agrupados y se toma la máxima distancia, formando una nueva clase con aquél vector que la maximiza como representante.

Paso 3.- Se distribuyen los restantes vectores en alguna de las clases existentes, anotando el vector *V* que esté a mayor distancia de su respectivo representante.

Paso 4.- Si la distancia de *V* a su representante es superior al valor umbral *h* se crea una clase con ese vector como representante y se vuelve al paso 3. En otro caso se termina.

5.4.3 Algoritmo de las k-medias

El *algoritmo k-medias* permite determinar la posición de *k* centroides que distribuyan de manera equitativa un conjunto de patrones. Debe notarse que, a diferencia de los algoritmos anteriores, este algoritmo tiene la particularidad de necesitar conocer a priori el número *k* de clases existentes.

La Figura 91 muestra un ejemplo de aplicación del algoritmo de *k*-medias. Inicialmente no se conoce qué patrones pertenecen a cada una de las clases, pero se sabe que la muestra está dividida en dos clases, por lo que se aplica el algoritmo de *k*-medias con *k*=2. En el instante *t*=1 se toman como centroides dos patrones al azar. En la figura se muestra en gris la zona en la que están los patrones que están mas cerca de ese centroide. Serán los patrones de cada una de las zonas los que se usen para calcular el nuevo centroide en la siguiente iteración. En los instantes sucesivos se aprecia como los centroides van viajando hasta su ubicación definitiva. En la tercera iteración los centroides alcanzan una posición estable que no cambiará en sucesivas iteraciones, por lo que el algoritmo finaliza.

Este algoritmo goza de una amplia difusión debido a que es simple, a que suele ofrecer unos resultados fiables, y a que no depende de ningún umbral heurístico. Tiene la desventaja de que es preciso conocer de antemano el número de clases en que se divide la muestra. Aunque esto, en las ocasiones en que se dispone de tal dato, es más una ventaja que un inconveniente. También debe

observarse que la solución obtenida puede ser dependiente de la elección que se haga de los puntos iniciales, por lo que debe ponerse cuidado en este punto. Esta variación del resultado dependiendo de los puntos iniciales ocurre sobre todo cuando la distinción entre clases no está muy clara, o cuando no se estima adecuadamente el número de clases.

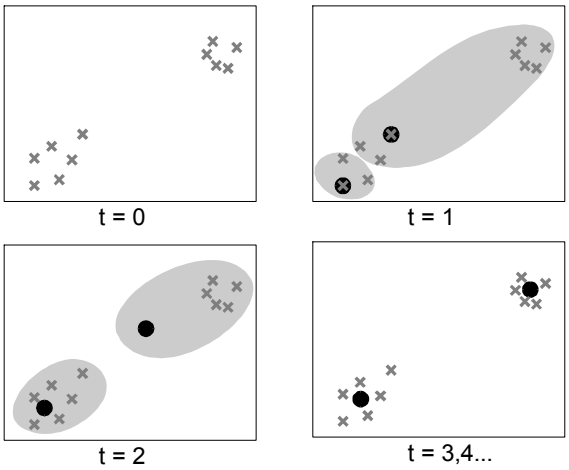


Figura 91.- Ilustración de la posición de los centroides (puntos negros) al aplicar el algoritmo de k-medias (con k=2), durante varias iteraciones para una muestra concreta.

- Algoritmo k Medias -

Paso 1.- Se hace t = 1. Se toma al azar k vectores de los P existentes y se convierten en centroides de cada una de las k clases respectivamente.

$$Z_1(1) \text{ de } a_1$$
$$\ldots$$
$$Z_k(1) \text{ de } a_k$$

Paso 2.- Se distribuyen las P-k muestras restantes entre las k clases. Se asigna cada vector a la clase que esté más próxima.

$$x \in a_j \Leftrightarrow |x\text{-}z_j(t)| < |x\text{-}z_i(t)| \quad \forall i = 1,2,\ldots k \,/\, i \neq j$$

Paso 3.- Se calcula los centroides de las clases como la media ponderada de los vectores de cada clase.

Paso 4.- Si alguno de los k centroides $Z_k(t)$ es distinto de los nuevos centroides $Z_k(t+1)$ hacer $t = t+1$ e ir al paso 2, en otro caso finalizar el algoritmo.

5.5. Conclusiones al capítulo

El resultado de la etapa de clasificación suele corresponder al último objetivo de un sistema de visión artificial. Por ejemplo, es aquí donde un sistema reconocimiento de caracteres clasifica una imagen como una letra determinada, o un sistema de reconocimiento biométrico identifica la imagen de un individuo como tal o cual persona. Es por tanto importante no apresurarse en esta última etapa.

Se ha visto que al clasificar un patrón se puede obtener un valor sobre la confianza de tal clasificación. Este valor puede corresponder incluso a una probabilidad para algunos tipos de clasificadores. Es el momento de hacer notar que, en la mayoría de las ocasiones, si la confianza en el resultado no es alta puede volverse a etapas anteriores (filtrado o segmentación) para, aplicando otros enfoques, obtener un resultado de mayor confianza.

La experiencia de los autores en los sistemas de visión artificial desarrollados demuestra que el usuario final suele ser poco comprensivo con los errores de la máquina en la clasificación, sobre todo en aquellos casos en los que el error es difícilmente justificable desde el punto de vista humano. Por ello casi siempre es mejor aplicar diferentes enfoques de preprocesamiento, segmentación y clasificación y obtener tantos resultados independientes como se pueda. Estos resultados pueden combinarse a su vez mediante otros clasificadores o mediante técnicas de votación[24]. En el peor de los casos se puede decidir que el sistema, ante un patrón dudoso, no ofrezca resultado, a fin de que ofrezca fiabilidad a los usuarios finales.

5.6. Bibliografía del capítulo

[GW93] caps. 9,

[24] Entre estas técnicas se pueden citar el algoritmo del elemento más votado, el algoritmo de la votación promediada y el algoritmo BKS. El primero devuelve el resultado del clasificador que más se repita. El segundo devuelve el resultado del clasificador que promediado con los demás con cierto coeficiente obtenga un resultado más alto. El último utiliza los resultados de N clasificadores sobre una muestra de ensayo para construir una matriz de dimensión N que ofrece resultados para cada combinación de resultados de los N clasificadores.

[JKS95] cap. 15,

[Mar93] caps. 2, 4 y 6,

[Loo97] cap. 5

Capítulo 6 Introducción a la Visión Tridimensional

Cuando un búho acecha una presa realiza rápidos y precisos movimientos con la cabeza. Estos movimientos le permiten obtener distintas perspectivas de la escena que observa. El cerebro del búho usa estas diferentes perspectivas para calcular con precisión la distancia y la posición exacta en la que se encuentra su próximo bocado.

El obtener la estructura tridimensional de una escena ha sido, y sigue siendo, uno de los retos de la visión artificial. Quizás debido a que es la forma en que los seres vivos ven, el problema central que clásicamente se considera en visión tridimensional es el de la reconstrucción de una escena 3D a partir de una o varias proyecciones del mundo real sobre superficies fotosensibles. Este capítulo se dedica a la formulación más básica de este problema que se conoce como el método del par estereoscópico.

Existen multitud de enfoques alternativos que también permiten recuperar una escena tridimensional. Estos enfoques se introducen brevemente en el último punto de este tema.

6.1. Método del par estereoscópico

A continuación se revisa el modelo de cámara para con él tratar de reconstruir una escena 3D partir de una imagen digital (visión monocular). Se expondrán las ecuaciones que relacionan un punto de una escena y su correspondiente dentro de la imagen digital, así como el proceso de calibración necesario para estimar los parámetros que gobiernan estas ecuaciones.

Posteriormente, se explicará por qué no esa posible reconstruir una escena 3D a partir de una única imagen 2D, surgiendo la necesidad de varias proyecciones distintas de la misma escena para calcular, por triangulación, una representación tridimensional de la misma. Así, con el fin de eliminar las limitaciones que subyacen al enfoque monocular se planteará la visión estereoscópica o binocular, y se expondrán sus capacidades, ventajas e inconvenientes.

6.1.1 Visión monocular

La visión monocular emplea un único observador para capturar una escena. Los elementos básicos que se necesita contemplar en visión monocular son: el mundo real tridimensional que denominaremos *escena*, y una imagen digital plana de esa escena (normalmente obtenida con una cámara).

Así, el problema, en su formulación más simple, se plantea como el de averiguar las coordenadas en el mundo real tridimensional de un punto de un objeto partiendo de una imagen digital bidimensional de una escena que contiene ese objeto. En el siguiente punto se verá que este problema se resuelve estudiando la relación entre la imagen digital y la escena.

Relación entre la imagen digital y la escena

Formalmente, el problema que plantea la visión monocular, consiste en saber cuáles son las coordenadas (x, y, z) de cierto punto P, a partir de una imagen bidimensional I que corresponde a una perspectiva de la escena, en la que aparece ese punto P en el píxel (i, j). Las variables que se deben contemplar para resolver el problema son:

O.- Origen del sistema de referencia de la escena.

P.- Punto de la escena tridimensional.

u, *v*, *a*.- Sistema de referencia que define la posición de la cámara.

C.- Posición del foco de la cámara.

P'.- Punto de coordenadas (*i*, *j*), dentro de la imagen digital, que se corresponde al punto *P*. O de otra forma, proyección de *P* en la imagen digital.

Representando todos los elementos de este problema se obtiene la Figura 92, que se conoce como modelo *Pin-Hole* o *cónico*. Este modelo supone una simplificación del modelo de lente fina (ver capítulo 2) ya que, por un lado, hace coincidir la posición del plano de formación de la imagen con el foco de la lente, y por otro, reduce el tamaño de la lente a un punto.

El modelo Pin-Hole tiene la ventaja de su simplicidad, pero sólo es aplicable cuando la distancia a la que están los objetos es mucho mayor que la distancia focal.

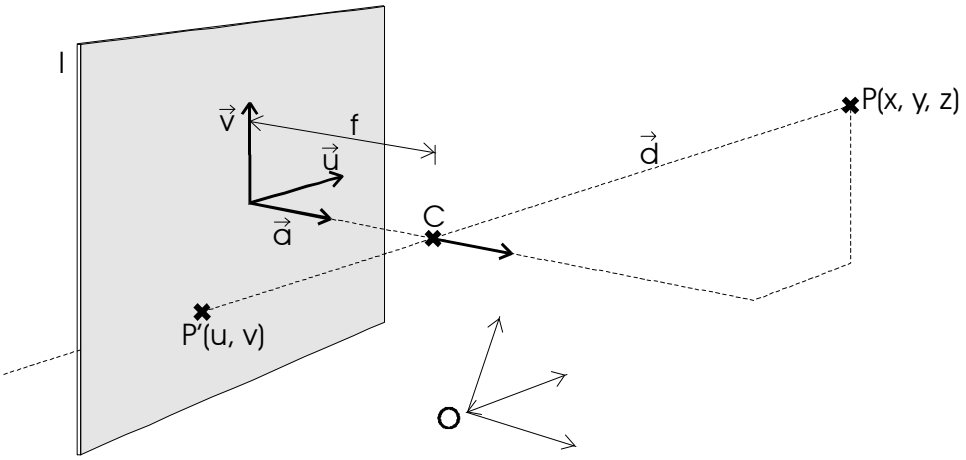


Figura 92.- Modelo Pin-Hole o perspectiva cónica. Obsérvese que todos los puntos de la recta *CP* se proyectan en el mismo punto *P'* de la imagen.

Observando la proyección sobre un plano vertical de la Figura 92 se obtiene la Figura 93. Sobre ésta se aprecia la proyección de *d* sobre *v* que resulta del producto escalar $d^{\wedge}v$ y la proyección de *d* sobre *a* que es $d^{\wedge}a$.

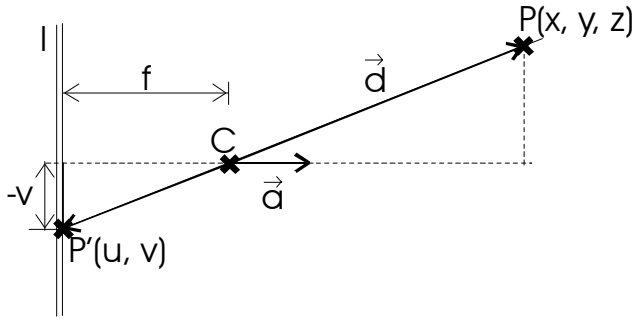


Figura 93.- Proyección lateral del modelo Pin-Hole.

Aplicando semejanza de triángulos se obtiene:

$$-\frac{v}{f}=\frac{\vec{d}\cdot\vec{v}}{\vec{d}\cdot\vec{a}}=\frac{\left|\vec{d}\right|\cdot\left|\vec{v}\right|\cdot\cos(\vec{d},\vec{v})}{\left|\vec{d}\right|\cdot\left|\vec{a}\right|\cdot\cos(\vec{d},\vec{a})}$$

Y como *v* y *a* son vectores unitarios se obtiene:

$$-\frac{v}{f}=\frac{\cos(\vec{d},\vec{v})}{\cos(\vec{d},\vec{a})}\Rightarrow -v=f\cdot\frac{\cos(\vec{d},\vec{v})}{\cos(\vec{d},\vec{a})}$$

Para la otra proyección, con un razonamiento similar se obtiene:

$$-u=f\cdot\frac{\cos(\vec{d},\vec{u})}{\cos(\vec{d},\vec{a})}$$

Sobre estas fórmulas se puede cambiar *d* por *P*-*C*, con lo que el problema estará expresado directamente en los parámetros que inicialmente se propusieron.

Por último, sólo resta transformar las coordenadas (*u*, *v*) del plano de formación de la imagen en coordenadas (*i* ,*j*) del plano digital de la imagen. Para hacer esta transformación se debe tener en cuenta la relación de tamaño de los píxeles dentro del plano de formación de la imagen, por lo que se llamará al ancho de un píxel *m*, y al alto *n*. También se deben conocer las coordenadas (i_0, j_0) del punto del plano digital de la imagen que se corresponden con las del centro del plano de formación de la imagen.

Cuando u vale 0, i vale i_0, y cuando i vale i_0+1 u toma valor igual al ancho de un píxel, es decir m. Aplicando un razonamiento similar para calcular la relación entre j y v se obtienen las igualdades:

$$u = m\,(i - i_0)$$

$$v = -\,n\,(j - j_0)$$

Sustituyendo finalmente se obtiene:

$$i = \frac{-f}{m} \cdot \frac{(\vec{P} - \vec{C}) \cdot \vec{u}}{(\vec{P} - \vec{C}) \cdot \vec{a}} + i_0 \qquad (6.1)$$

$$j = \frac{f}{n} \cdot \frac{(\vec{P} - \vec{C}) \cdot \vec{v}}{(\vec{P} - \vec{C}) \cdot \vec{a}} + j_0 \qquad (6.2)$$

Estas fórmulas permiten, a partir de las coordenadas (i, j) de un píxel en una imagen digital, conocer cuáles son los valores (x, y, z) posibles del punto P dentro de la escena tridimensional. Como sólo hay dos ecuaciones y tres incógnitas, no será posible conocer sus coordenadas sino la relación entre ellas y un parámetro. Así, si alguna de las tres coordenadas se conociese, automáticamente se obtendrían las otras dos (por ejemplo, si se sabe que un objeto está sobre el suelo y el suelo tiene coordenada $z = 0$).

6.1.2 Calibración

Las ecuaciones (6.1) y (6.2) permiten conocer la relación entre los píxeles de una imagen digital, y los puntos de los que son proyección en la escena tridimensional. No debe pasarse por alto, sin embargo, que es preciso obtener los valores de todos los parámetros que aparecen en estas ecuaciones. La calibración es el proceso que se encarga de determinar los valores de los parámetros que intervienen en el proceso de formación de la imagen. En este modelo la calibración consiste precisamente en obtener el valor de los parámetros de estas ecuaciones (ver Tabla 6).

Estos parámetros se pueden dividir en dos grandes grupos: los *intrínsecos* (interiores a la cámara), y los *extrínsecos* (los exteriores). Para determinar los valores de tales parámetros suele utilizarse una imagen de calibración que dispone

de N puntos de los que se conocen sus posiciones. Esto suele conseguirse recurriendo a una imagen de la escena en la que aparece una plantilla de calibración, que es un elemento formado por uno o varios paneles en los que aparecen N puntos con posiciones relativas conocidas (ver Figura 94).

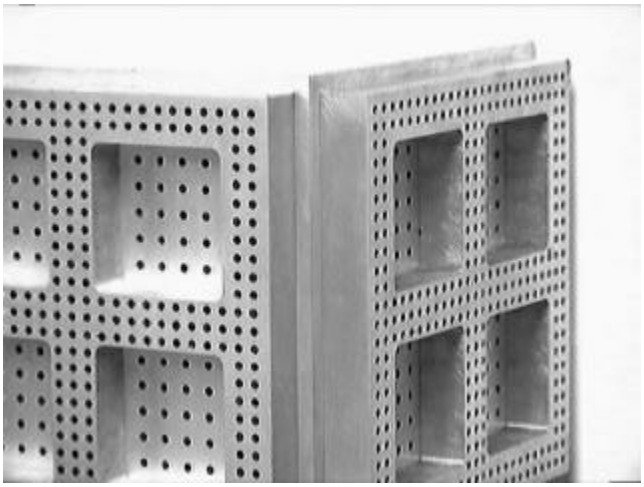


Figura 94.- Plantilla de calibración consistente en un cubo con N marcas.

Conociendo la posición de los N puntos de la plantilla de calibración se pueden plantear $2N$ ecuaciones, en ellas será posible despejar los parámetros desconocidos.

Extrínsecos	a, u, v	Orientación de la cámara respecto al sistema de referencia externo
	c	Posición de la cámara respecto al sistema de referencia externo
Intrínsecos	i_0, j_0	Punto focal
	$f/n, f/m$	Píxeles horizontales y verticales que ocupa la focal

Tabla 6.- Clasificación de los parámetros del modelo de visión monocular en función de su relación con el dispositivo de captura.

Cálculo matricial

En el siguiente apartado se realiza un desarrollo que llevará a una formulación que permitirá realizar el proceso de calibración de una manera sencilla.

En lo que sigue se utilizarán coordenadas homogéneas, es decir coordenadas a las que a las habituales *x*, *y*, *z* se añade una coordenada l (*con* $l \neq 0$) que facilita el cálculo matricial. Como $l \neq 0$ siempre es posible deshacer su intrusión.

Partiendo de (6.1) se desarrolla:

$$i = -\frac{f}{m} \cdot \frac{\vec{P} \cdot \vec{u} - \vec{C} \cdot \vec{u}}{(\vec{P} - \vec{C}) \cdot \vec{a}} + i_0 \;\Rightarrow\; i = -\frac{f}{m} \cdot \frac{\vec{u}^t \cdot \vec{P}}{(\vec{P} - \vec{C}) \cdot \vec{a}} + \frac{f}{m} \cdot \frac{\vec{u}^t \cdot \vec{C}}{(\vec{P} - \vec{C}) \cdot \vec{a}} + i_0$$

Igualmente partiendo de (6.2) se obtiene:

$$j = \frac{f}{n} \cdot \frac{\vec{v}^t \cdot \vec{P}}{(\vec{P} - \vec{C}) \cdot \vec{a}} - \frac{f}{n} \cdot \frac{\vec{v}^t \cdot \vec{C}}{(\vec{P} - \vec{C}) \cdot \vec{a}} + j_0$$

Llamando

$$l = (\vec{P} - \vec{C}) \cdot \vec{a} \;\Rightarrow\; l = \vec{a}^t \vec{P} - \vec{a}^t \vec{C}$$

se obtiene el sistema

$$\begin{cases} i\,l = -\dfrac{f}{m}\vec{u}^t \vec{P} + \dfrac{f}{m}\vec{u}^t \vec{C} + i_0\, l \\ j\,l = \dfrac{f}{n}\vec{v}^t \vec{P} - \dfrac{f}{n}\vec{v}^t \vec{C} + j_0\, l \\ l = \vec{a}^t \vec{P} - \vec{a}^t \vec{C} \end{cases}$$

que se puede escribir en forma matricial, poniendo atención a la matriz central que sólo contiene parámetros intrínsecos.

$$\begin{pmatrix} l \cdot i \\ l \cdot j \\ l \end{pmatrix} = \begin{pmatrix} \frac{-f}{m} & 0 & i_0 \\ 0 & \frac{f}{n} & j_0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} \vec{u}^t \vec{P} - \vec{u}^t \vec{C} \\ \vec{v}^t \vec{P} - \vec{v}^t \vec{C} \\ \vec{a}^t \vec{P} - \vec{a}^t \vec{C} \end{pmatrix}$$

Desarrollando la matriz de la derecha, y poniendo *P* como un producto externo, se obtiene:

$$\begin{pmatrix} l \cdot i \\ l \cdot j \\ l \end{pmatrix} = \begin{pmatrix} \frac{-f}{m} & 0 & i_0 \\ 0 & \frac{f}{n} & j_0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} u_x & u_y & u_z & -\vec{u}^t \vec{C} \\ v_x & v_y & v_z & -\vec{v}^t \vec{C} \\ a_x & a_y & a_z & -\vec{a}^t \vec{C} \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \Rightarrow$$

El uso de notación fasorial para la 3ª matriz permite observar fácilmente que contiene dos componentes. La primera, que contiene sólo parámetros intrínsecos, determina la orientación de la cámara en la escena. La segunda se refiere a la posición del origen de coordenadas de la escena respecto al sistema de referencia de la cámara que, recordemos, está en *C*.

$$\begin{pmatrix} l \cdot i \\ l \cdot j \\ l \end{pmatrix} = \begin{pmatrix} \frac{-f}{m} & 0 & i_0 \\ 0 & \frac{f}{n} & j_0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \left(\begin{array}{ccc|c} & \vec{u} & & -\vec{u}^t \vec{C} \\ & \vec{v} & & -\vec{v}^t \vec{C} \\ & \vec{a} & & -\vec{a}^t \vec{C} \end{array}\right) \cdot \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

Llamando a la matriz de intrínsecos *K*, a la orientación de la cámara en la escena *R*, y a la posición del origen de referencia de la escena respecto al sistema de referencia de la cámara *t*, se obtiene:

$$\begin{pmatrix} l \cdot i \\ l \cdot j \\ l \end{pmatrix} = K \cdot \left[R \mid \vec{t} \right] \cdot \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

En la mayoría de las ocasiones[25] es posible transformar el sistema anterior en uno de la forma:

$$\begin{pmatrix} l \cdot i \\ l \cdot j \\ l \end{pmatrix} = \begin{pmatrix} w_{1,1} & w_{1,2} & w_{1,3} & w_{1,4} \\ w_{2,1} & w_{2,2} & w_{2,3} & w_{2,4} \\ w_{3,1} & w_{3,2} & w_{3,3} & w_{3,4} \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

donde la matriz *W* se denomina matriz de proyección perspectiva, y es tal que aplicada a un punto del espacio tridimensional proporciona su proyección en la imagen.

Aquí, el proceso de calibración consiste en calcular los valores de las componentes de esta matriz. Desarrollando la expresión anterior, despejando l y sustituyendo en las dos primeras ecuaciones, se obtiene el siguiente sistema de ecuaciones de 12 incógnitas:

[25] Esto no es posible cuando el ángulo que forma el plano de la imagen con el eje axial de la cámara es distinto de cero. En estos casos es posible tal cambio si se introduce un parámetro δ en la posición (1,2) de la matriz de intrínsecos.

$$\begin{pmatrix} x & y & z & 1 & 0 & 0 & 0 & 0 & -xi & -yi & -zi & -i \\ 0 & 0 & 0 & 0 & x & y & z & 1 & -xj & -yj & -zj & -j \end{pmatrix} \cdot \begin{pmatrix} w_{1,1} \\ w_{1,2} \\ w_{1,3} \\ w_{1,4} \\ w_{2,1} \\ w_{2,2} \\ w_{2,3} \\ w_{2,4} \\ w_{3,1} \\ w_{3,2} \\ w_{3,3} \\ w_{3,4} \end{pmatrix} = 0$$

O más concisamente:

$$\begin{pmatrix} x & y & z & 1 & 0 & 0 & 0 & 0 & -xi & -yi & -zi & -i \\ 0 & 0 & 0 & 0 & x & y & z & 1 & -xj & -yj & -zj & -j \end{pmatrix} \cdot W = 0$$

Así, conociendo las coordenadas (x, y, z) del espacio tridimensional de un punto P, y las coordenadas (i, j) de su proyección correspondiente sobre el plano digital se tienen 2 ecuaciones con las 12 componentes de la matriz W como incógnitas. Se aprecia que para cumplir el objetivo de determinar completamente la matriz W, se necesitarán al menos 6 puntos, para tener 12 ecuaciones con las que determinar los 12 parámetros de W. Además, estos 6 puntos no deberán estar todos en el mismo plano, o el sistema resultará indeterminado[26].

[26] Puesto que se trabaja con coordenadas homogéneas la escala es irrelevante. En este caso es posible establecer por ejemplo el valor de $w_{3,4}$ a 1 y el resultado no cambiará. Así, el sistema tendrá 12 ecuaciones y 11 incógnitas y estará sobredeterminado. En este caso puede utilizarse algoritmos de minimización del error, como el método de mínimos cuadrados, para estimar W en vez de proceder a su cálculo directo.

6.1.3 Visión estereoscópica

A partir de las coordenadas (i, j) de un punto P de una imagen obtenida con una cámara calibrada no se pueden calcular las coordenadas (x, y, z) en las que un punto P está situado dentro de la escena. Esto se debe a que sólo se dispone de 2 ecuaciones y sin embargo existen 3 incógnitas (ver ecuaciones (6.1) y (6.2)). La utilización de varias imágenes (con diferentes perspectivas del objeto) proporciona una vía para resolver este problema. Con dos imágenes diferentes en las que aparezca P se tendrán 4 ecuaciones y sólo 3 incógnitas, por lo que el problema podría resolverse.

El problema de correspondencia

Sin embargo, el uso de más de una imagen plantea un problema que se conoce como el problema de *correspondencia*. Este problema se plantea al intentar hacer corresponder los puntos de 2 imágenes digitales distintas. Es decir, conociendo las coordenadas (i_1, j_1) de un píxel dentro de la imagen digital I_1 que corresponde a un punto P de una escena tridimensional, cuáles son las coordenadas (i_2, j_2) correspondientes a la representación del mismo punto P en la imagen digital I_2. Este problema se resuelve encontrando la *homografía* entre los puntos P_1 y P_2. Siendo una homografía una transformación que relaciona las diferentes proyecciones de un punto.

Restricción epipolar

Existe una propiedad geométrica, que posee todo sistema de visión estéreo, que ayudará a resolver el problema de correspondencia. Esta propiedad se conoce como la restricción epipolar.

La *restricción epipolar* dice que dado un píxel P_1 sobre el plano de imagen I_1, su correspondencia, en el plano de imagen I_2, debe estar dentro de la recta R_2 que resulta de la intersección del plano epipolar con la imagen I_2. Siendo el plano epipolar el que definen los tres puntos C_1,C_2 y P (ver Figura 95).

Dado un punto P del mundo real tridimensional y dadas sus proyecciones P_1 y P_2 en dos imágenes que ofrecen distintas perspectivas se obtienen las siguientes ecuaciones:

$$l_{\ 1} \cdot P_1 = K_1 \cdot \left[R_1 \mid \vec{t}_1\right] \cdot \vec{P}$$

$$l_{\ 2} \cdot P_2 = K_2 \cdot \left[R_2 \mid \vec{t}_2\right] \cdot \vec{P}$$

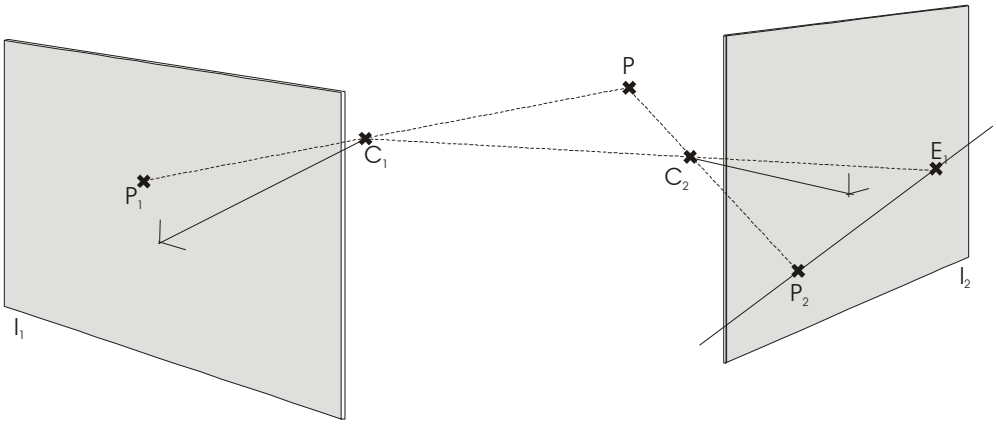


Figura 95.- Representación del problema de correspondencia en la visión estéreo. El punto P tiene una representación P_1 en el plano de la imagen I_1 y otra distinta P_2 en el plano de la imagen I_2. La recta epipolar es la que definen los puntos E_1 y P_2.

Si por simplicidad se supone que el sistema de referencia coincide con el de I_1 la primera ecuación se transforma en:

$$l_{\ 1} \cdot P_1 = K_1 \cdot \left[I \mid 0\right] \cdot \vec{P} \ \Rightarrow$$

$$l_{\ 1} \cdot P_1 = K_1 \cdot \vec{P} \tag{6.3}$$

y la segunda ecuación cambia a:

$$l_{\ 2} \cdot P_2 = K_2 \cdot R_2 \cdot \vec{P} + K_2 \cdot \vec{t}_2$$

Usando el resultado (6.3) sobre la fórmula anterior se obtiene:

$$l_{\ 2} \cdot P_2 = K_2 \cdot R_2 \cdot K_1^{-1} \cdot l_{\ 1} \cdot P_1 + K_2 \cdot \vec{t}_2 \ \Rightarrow$$

$$l_2 \cdot P_2 = l_1 \cdot \underbrace{K_2 \cdot R_2 \cdot K_1^{-1}}_{H_\infty} \cdot P_1 + \underbrace{K_2 \cdot \vec{t}_2}_{e} \tag{6.4}$$

que suele escribirse:

$$l_2 \cdot P_2 = l_1 \cdot H_\infty \cdot P_1 + e$$

El parámetro H_∞ se conoce como la homografía del infinito, ya que relaciona las proyecciones de un punto que se encuentre en el infinito. Claramente esta homografía sólo tiene en cuenta K y R, pues la posición del observador, representada por t, no influye cuando lo observado se encuentra en el infinito. En la práctica basta con que la distancia al punto sea mucho mayor que las distancias focales. Por ejemplo, cuando se mira montañas que se encuentran en el horizonte la posición en la que se encuentra el observador es indiferente respecto al punto de la retina en el que se proyectan las montañas.

Multiplicando escalarmente por K_2^{-1} a ambos lados de la ecuación:

$$l_2 \cdot K_2^{-1} \cdot P_2 = l_1 \cdot R_2 \cdot K_1^{-1} \cdot P_1 + \vec{t}_2$$

Multiplicando vectorialmente por t_2, y sabiendo que $t_2 \times t_2 = 0$ se obtiene:

$$l_2 \cdot t \times \left[K_2^{-1} \cdot P_2\right] = l_1 \cdot t \times R_2 \cdot K_1^{-1} \cdot P_1$$

Multiplicando por $\left[K_2^{-1} \cdot P_2\right]$ que es perpendicular a $t \times \left[K_2^{-1} \cdot P_2\right]$ se obtiene:

$$0 = l_1 \cdot \left[K_2^{-1} \cdot P_2\right] \cdot t \times R_2 \cdot K_1^{-1} \cdot P_1 \;\Rightarrow$$

$$0 = P_2 \cdot \underbrace{K_2^{-1} \cdot t \times R_2 \cdot K_1^{-1}}_{F} \cdot P_1 \;\Rightarrow$$

que agrupando queda:

$$0 = P_2 \cdot F \cdot P_1 \tag{6.5}$$

Siendo F una matriz, con determinante distinto de cero, conocida como *matriz fundamental*, que relaciona puntos de la imagen I_1 con los de la imagen I_2.

Cálculo de la matriz fundamental

Si ambas cámaras están calibradas se conocen todos los parámetros del siguiente sistema, con lo que el cálculo de P_1 y P_2 se puede abordar para los puntos de calibración.

$$l_1 \cdot P_1 = K_1 \cdot \left[R_1 \mid \vec{t}_1\right] \cdot \vec{P}$$

$$l_2 \cdot P_2 = K_2 \cdot \left[R_2 \mid \vec{t}_2\right] \cdot \vec{P}$$

Conociendo las coordenadas de suficientes puntos P_1 y P_2 se puede determinar la matriz F, planteando ecuaciones iguales a (6.5) sobre los puntos de calibración. Una vez conocida la matriz F es posible, a partir de un punto P_1 determinar cuál es su correspondiente P_2, con lo que se salvaría el problema de correspondencia.

6.1.4 Conclusiones al problema de visión estéreoscópica

Se ha visto cómo se puede obtener información tridimensional de una escena (concretamente puntos 3D de la superficie de los objetos que contiene) mediante visión estereoscópica. Sin embargo la realidad es mas compleja que el modelo que se ha presentado, en este punto se discuten algunos de los problemas que surgen cuando se construyen sistemas de visión 3D de este tipo.

La metodología general para abordar el problema pasa por la selección de puntos en una imagen e identificación de estos en la otra imagen (problema de la correspondencia), así como la posterior obtención de la profundidad, para lo que es imprescindible contar con una precisa calibración de las cámaras usadas. La calibración de las cámaras se complica cuando éstas deben moverse (p.e. si están sobre un robot móvil).

El problema de la correspondencia es el más crítico en los sistemas de visión estéreo, y aunque se pueden encontrar múltiples propuestas que intentan resolverlo ninguna da una solución a su totalidad debido a problemas como los que se detallan en el siguiente apartado.

Problemas relacionados al problema de la correspondencia

El primer problema estriba en que es necesario que los puntos cuyas coordenadas 3D se desean calcular mediante visión estéreo estén visibles en las dos imágenes. Sin embargo, las dos imágenes no ven la misma porción de la escena, y puntos de una imagen pueden no estar visibles en la segunda imagen (por estar ocluidos o por no pertenecer al campo de visión de la segunda cámara) lo que dificulta el establecimiento de correspondencias originando el problema de *oclusión*. Para minimizar este problema se pueden añadir más observadores (más cámaras) de tal manera que puntos que no se vean desde una cámara puedan ser captados por otras. Esta solución complicará la calibración del sistema.

Además, es posible que el sistema encuentre *falsas correspondencias*, al tener que elegir entre varios candidatos en una imagen a ser correspondencia de un punto en otra imagen. Este inconveniente se reduce mediante el empleo de restricciones como la restricción epipolar.

En la obtención de puntos de las imágenes también se presenta un problema conocido como el problema de la *repetitividad*. Se debe a que un rasgo de una imagen se compone de varios píxeles, por lo que hay que decidir uno concreto entre ellos a la hora de seleccionarlo. Esta limitación se presenta tanto en el proceso de selección de correspondencias de puntos como en el de selección de puntos en la etapa de calibración. El efecto negativo del posible error cometido en la selección de puntos para calibración puede reducirse si se usan muchos más puntos que los estrictamente necesarios.

Por otra parte, los modelos de las cámaras aproximan el funcionamiento de las mismas en el paso de puntos 3D a puntos 2D proyectados en la imagen. Sin embargo, existen muchos parámetros que determinan el proceso de formación de la imagen y no se contemplan todos en el modelo de lente fina que se ha planteado. Existen modelos de cámaras en los que intervienen más parámetros, para modelizar el proceso de formación de la imagen de una forma más realista. El más popular es el *modelo de Tsai*, en el que intervienen del orden de 21 parámetros, entre los que se puede destacar los relativos a la distorsión radial de la lente. Este modelo es no lineal, a diferencia del modelo Pin-Hole comentado anteriormente que sí lo era.

Se podrían citar muchos más problemas (iluminación de la escena, texturas de las superficies de los objetos, etc.) que imposibilitan actualmente el uso de sistemas de visión estéreo en problemas generales, restringiendo su uso a

escenarios específicos y muy controlados. Debido a estos problemas actualmente existe una gran atención en la investigación de nuevos enfoques.

6.2. Otros enfoques para la visión 3D

Los métodos de reconstrucción 3D clásicamente se pueden clasificar en dos grupos: los *métodos activos* y los *métodos pasivos*. Los activos son los que exigen un observador que debe realizar alguna operación o cumplir algún requisito, o que necesitan de una interferencia sobre la escena que se observa. Los pasivos son los que no precisan de ninguno de estos elementos basándose únicamente en imágenes de intensidad para reconstruir la triangulación.

Así por ejemplo son métodos activos: aquéllos en los que se proyectan haces controlados de energía (luz o sonido sobre la escena) desde una posición y orientación conocida, o los que utilizan un observador activo (aquél que se encuentra implicado en algún tipo de actividad encaminada a controlar la geometría utilizada para la discretización). Como método pasivo se puede citar el que se ha descrito en este capítulo basado en visión estereoscópica.

6.2.1 Ejemplos de otros enfoques

Entre los métodos activos se pueden citar los métodos basados en la proyección de luz estructurada, los de *tiempo de vuelo*, el *método de Moire*, *inferometría holográfica*, etc. La distinción entre métodos activos y pasivos no es clara, así el proceso de calibración puede hacer que el método estereoscópico que se ha explicado se pueda considerar un método activo. Algunos ejemplos de métodos puramente pasivos son los métodos de cálculo a partir del sombreado, intersección de volúmenes, secuencia de imágenes y visión estereoscópica sin calibrar. En los siguientes puntos se realiza un descripción superficial de algunos de estos métodos.

Tiempo de vuelo

Método activo en el que se mide el tiempo invertido por una señal de velocidad conocida en recorrer la distancia que separa una región de la escena del dispositivo emisor-receptor. La señal suele ser de naturaleza acústica o electromagnética.

Plantean problemas relacionados con la baja energía reflejada. Es por ello que son usados principalmente sobre objetos cercanos (para tareas de modelado 3D por ejemplo). Además los escáneres 3D son caros y voluminosos en comparación con las cámaras convencionales.

Inferometría de Moiré

El motivo de interferencia obtenido al iluminar y observar una escena a través de sendos dispositivos idénticos de tipo rejilla permite la recuperación de la información tridimensional. Se basa en que la superposición de dos motivos de la misma frecuencia espacial produce una interferencia de baja frecuencia que únicamente varía con la diferencia de fase. Los inconvenientes son que los gradientes de la superficie del objeto han de ser acotados, y ha de existir continuidad. La Figura 96 muestra la disposición de un sistema de este tipo.

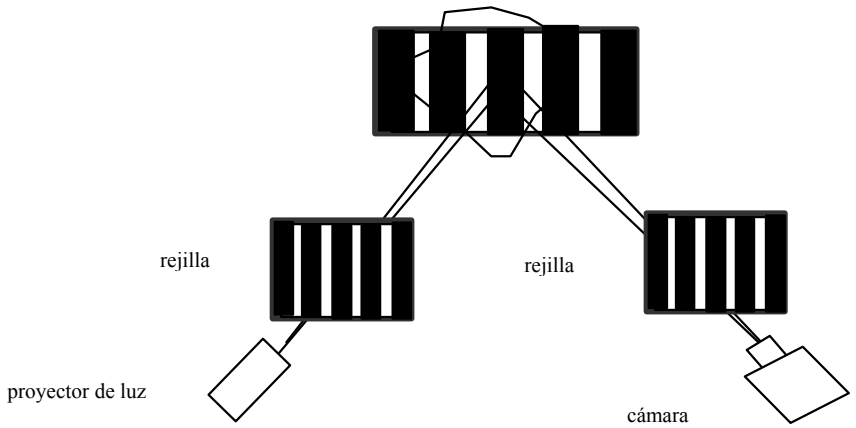


Figura 96.- Disposición de un sistema de adquisición de datos 3D basado en el método de Interferometría de Moiré.

Proyección de luz estructurada

La obtención de los puntos 3D se realiza mediante triangulación activa, empleando una cámara de imágenes intensidad y un proyector de luz de dirección controlada. El proyector de elevada potencia lumínica origina fuertes gradientes de luminancia en zonas arbitrarias de la escena. Para muestrear toda la escena ha de haber un

mecanismo de barrido que asegure la iluminación secuencial. Este método proporciona mapas tridimensionales densos y precisos, estando hoy en día muchos sensores de rango basados en esta técnica.

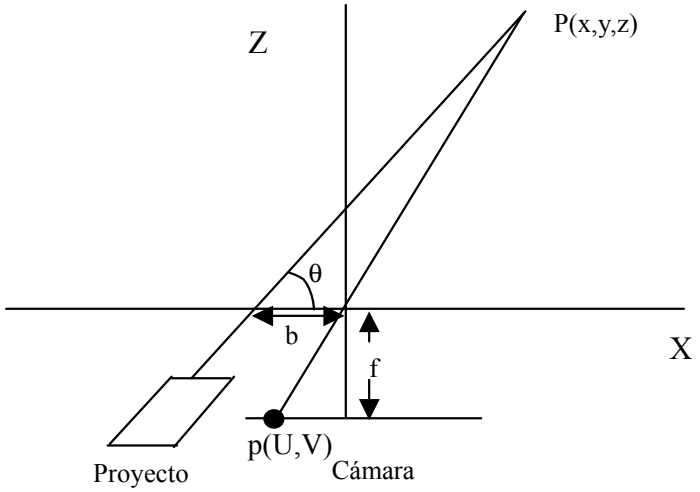


Figura 97.- Disposición de un sistema de visión 3D mediante proyección de luz estructurada. La obtención de puntos 3D se basa en el método de triangulación activa: conocidos *b*, *f*, *U* y *V*, es posible obtener las coordenadas 3D de *P*.

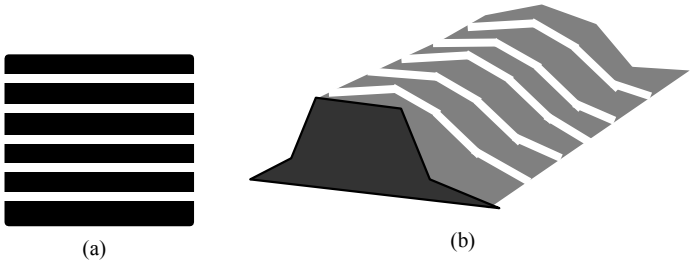


Figura 98.- .- (a) Patrón de luz estructurada utilizado para la proyección de luz estructrurada, y (b) objeto reflejando ese patrón de luz. Los puntos iluminados se pueden obtener mediante triangulación activa.

6.2.2 Imágenes de rango

La representación digital de estas imágenes 3D se realiza mediante las *imágenes de rango*. Las imágenes de rango son una clase especial de imágenes digitales. Cada píxel de una imagen de rango expresa la distancia entre un sistema de referencia conocido y un punto visible en la escena. Una imagen de rango reproduce la estructura 3D de una escena mediante una representación realista de la superficie muestreada.

Las imágenes de rango pueden venir dadas, bien mediante una lista de coordenadas 3D de puntos sin especificaciones de orden ni conectividad entre ellos, como se muestra en la Figura 99 (a esta forma de representación se la denomina nube de puntos), o bien mediante una matriz de valores de profundidad de puntos a lo largo de las direcciones *x* e *y* de la imagen. Las imágenes que vienen dadas de esta forma se denominan *imágenes de profundidad*, *mapas de profundidad*, *perfiles de superficie*, o *imágenes* $2^{1/2}D$.

Figura 99.- Nube de puntos de la superficie de un cráneo humano adquirida con un digitalizador manual (táctil).

Las imágenes de rango constituidas por nubes de puntos requieren una etapa de aproximación de esa nube de puntos a una superficie matemática. La problemática de la elección del tipo de malla (triangular, cuadrangular...), la forma de aproximar la malla (interpolación mediante triángulos de Bezier, triangulación

de Delaunay, etc...), la eliminación de puntos no significativos para acelerar los cálculos (usando por ejemplo técnicas de erosión 3D) son aún problemas abiertos.

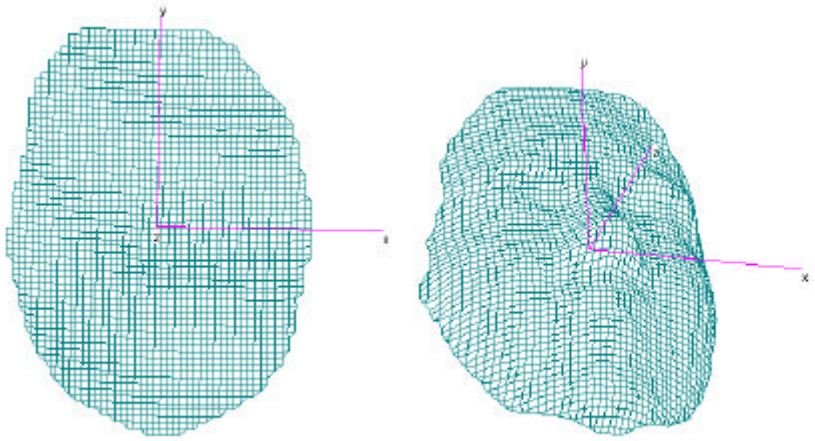


Figura 100.- Imagen de rango adquirida mediante un sensor de rango 3D láser que emplea triangulación activa para el cálculo de las coordenadas 3D. La imagen de la derecha presenta la vista frontal del objeto y permite apreciar la resolución del escáner. La de la izquierda esta rotada para que pueda apreciarse su forma.

Los mapas de profundidad suelen denominarse 2½D. Para construir representaciones 3D completas, son necesarios varios mapas de proyección y un método de integración entre distintos mapas. En este proceso de integración surge nuevamente el problema de correspondencia, conocido en este ámbito como *problema de registro*.

6.3. Conclusiones al capítulo

Los métodos activos se están mostrando útiles en problemas industriales donde la interferencia con la escena es posible. Sin embargo, en muchos casos esta interferencia es inadmisible, por ejemplo en ambientes exteriores. Por ello los métodos pasivos son más fáciles de integrar en problemas reales.

Actualmente se están estudiando otros métodos de recuperación de la forma tridimensional como aquéllos que actúan a partir de la sombra y aquellos que lo hacen a partir de la textura.

En cualquier caso este tema, como otros que han sido descritos en este libro, constituye un problema abierto que seguro encontrará nuevas soluciones en los próximos años.

6.4. Bibliografía del capítulo

[JKS95] cap. 12,

[TV98] caps. 6, 7 y 10,

[Gon00] caps. 9 y 10

Anexo A Clasificación con el perceptrón multicapa

El sistema nervioso se encarga de recoger los impulsos del mundo que nos rodea y de coordinar y dirigir todas las actividades de los órganos de acuerdo a lo que ha percibido de ese exterior. Este complejo sistema tiene como unidad funcional un único tipo de células: las neuronas. Las neuronas disponen de un elemento que se llaman *axón* que permite trasmitir a otras neuronas a través de las *dendritas* impulsos eléctricos de diferente intensidad. Estas células se disponen en forma de complejas redes y mediante unos procesos, conocidos como *procesos sinápticos*, según los cuales se excitación o se inhiben unas a otras, son las responsables de las capacidades de aprendizaje y comprensión que caracterizan a los seres vivos que las poseen.

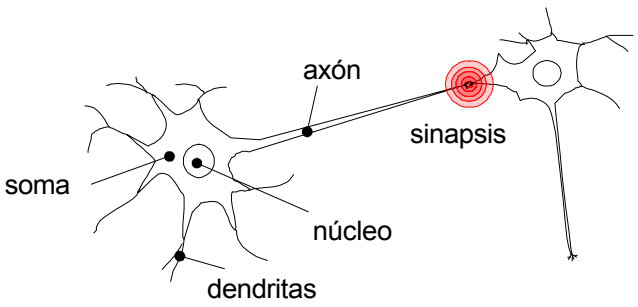


Figura 101.- Modelo biológico que representa la conexión entre dos neuronas.

En un intento de imitar el funcionamiento de estas estructuras naturales surgen las *redes de neuronas artificiales*. Reproducir el sistema nervioso aún está lejos de nuestro alcance, pero aún así, se han creado unos modelos que, aunque mucho más simples en su proceso que los modelos biológicos actuales, se muestran extremadamente útiles en problemas de clasificación. En este anexo se aborda un tipo especial de redes de neuronas artificiales, que se conocen como *perceptrones*, que pese a su simplicidad se adecuan de manera notable a los problemas de reconocimiento de patrones.

A.1. Introducción a las redes de neuronas artificiales

El funcionamiento de las redes de neuronas artificiales tiene su base en la interacción de unos elementos, las *células nerviosas* o *neuronas*, a través de unas conexiones llamadas *conexiones sinápticas*. Tales conexiones tienen su origen en una neurona (a la que se denomina neurona de salida) y su destino en otra neurona (a la que se llama neurona de entrada), tienen asociadas además estas conexiones un *peso* que es un valor real variable que determina la influencia de la misma. Una misma neurona puede ser destino de varias conexiones sinápticas. Por ello, las entradas que una neurona recibe se procesan, mediante la *función de activación*, para definir el *estado de activación* de la célula nerviosa. A partir de la función de activación se genera la salida de la neurona mediante la *función de salida*[27].

Se puede distinguir tres tipos principales de neuronas artificiales (ver Figura 102). Las *neuronas de entrada*, son aquéllas por donde se introducen los datos, a partir de los cuales la red producirá la respuesta. Son las únicas neuronas cuyo estado de activación se impone directamente desde el exterior. Las *neuronas de salida* son las que con su estado de activación definen la respuesta de la red ante el estímulo suministrado a su entrada. Puede haber, por último, una serie de *neuronas intermedias*, llamadas *neuronas ocultas*, que procesan y propagan información desde la entrada hasta la salida. Como se puede apreciar, por esta descripción que se ha realizado de las redes de neuronas, la salida de una red sólo

[27] En su definición más general se definen tres funciones asociadas a cada neurona. La función de entrada, la función de activación y la función de salida, aunque como la función de salida suele ser la función identidad en general no se tiene en cuenta. Hay autores que denominan *función de transferencia* al conjunto formado por la función de activación y la función de salida.

depende: de la entrada, de las conexiones que existan entre las células y de los pesos que éstas tengan asociados.

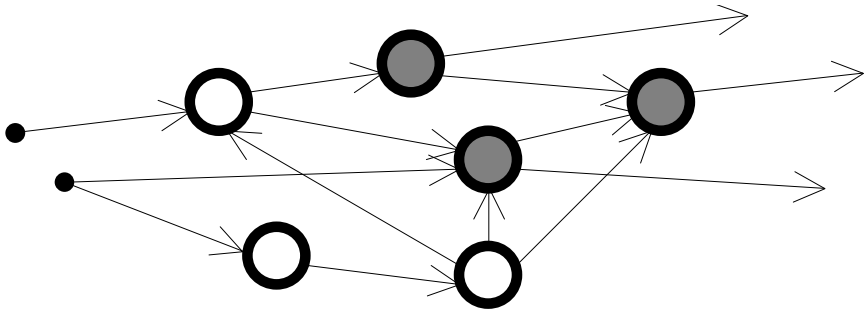


Figura 102.- Ejemplo de una red de neuronas genérica. Se presentan en negro y más pequeñas las neuronas de entrada y, sombreadas las de salida. El resto son neuronas intermedias u ocultas.

A.1.1 El proceso de aprendizaje de una red

El cálculo de los pesos que deben poseer las conexiones de una red de neuronas para que realice la función que se desea se conoce como *aprendizaje*. En redes triviales el ajuste de estos pesos se puede hacer manualmente, pero la tarea se vuelve imposible en redes de una complejidad mínima. Se han inventado diferentes algoritmos que realizan esta tarea de manera más o menos automática sobre distintos tipos de redes, aunque todos tienen inconvenientes[28].

Desde un punto de vista puramente matemático, dejando de lado lo que las entradas o las salidas de la red significan en este contexto, el problema consiste en encontrar el valor de unas incógnitas (los pesos) para que unas funciones (las ecuaciones de las neuronas) obtengan ciertos valores (la salida deseada) cuando se dan unos valores determinados a sus parámetros (las entradas a la red). En caso de no poder obtenerse exactamente el valor buscado se podría pensar en minimizar el error entre la salida deseada y la obtenida. Esto lleva al problema de encontrar el mínimo global de una función cualquiera, problema para el que en general no se conoce solución. Son aplicables sin embargo una infinidad de procedimientos

[28] El proceso de aprendizaje de un conjunto de neuronas reales constituye un tema por descubrir para la biología actual.

numéricos, basados generalmente en métodos del gradiente, que desgraciadamente sólo permiten obtener soluciones locales a este problema. El algoritmo más popular es el de *aprendizaje por retropropagación*.

Propiedad de generalización de las redes de neuronas

Durante el entrenamiento las salidas de una red de neuronas se aproximan a ciertos valores objetivo. Esto puede bastar para ciertas aplicaciones, pero el mayor interés despertado por las redes de neuronas se debe a su capacidad para *generalizar*. Se llama generalizar a la propiedad que le permite generar salidas adecuadas frente a entradas que no se encuentran en el conjunto de entrenamiento y que por tanto la red no ha aprendido previamente. La generalización no siempre es posible. Se destacan tres condiciones necesarias para que pueda conseguirse generalización:

- Que las entradas contengan suficiente información para que se pueda alcanzar las salidas deseadas al menos en el entrenamiento. Si la red no es capaz de aprender algo difícilmente será capaz de generalizar posteriormente.

- Que la función a interpolar sea, en cierto sentido, suave. Es decir, que un pequeño cambio en la entrada debe producir un pequeño cambio en la salida. Esto implica que la función que se emula sea continua hasta la primera derivada. Por ello las salidas de funciones como los generadores de números pseudo-aleatorios o algoritmos como los de encriptación no pueden ser emuladas con una red de neuronas.

- Que el conjunto de entrenamiento sea suficientemente grande. Esto se debe a que la generalización se produce en dos sentidos: interpolando y extrapolando. La interpolación en general da buenos resultados, pero la extrapolación no, por ello se deben tener suficientes casos de entrenamiento de manera que se que evite la extrapolación.

Conjuntos necesarios para entrenar una red

Para el ajuste de los pesos de las redes de neuronas, se precisan unos patrones con las entradas a la red y otros con las salidas deseadas frente a esos estímulos, constituyendo lo que se llama *conjunto de entrenamiento* (CE). Se usará este

conjunto para entrenar la red, mediante un procedimiento iterativo, que consiste en variar los pesos buscando minimizar la función de error que se haya determinado.

Clásicamente se distingue otro conjunto, denominado *conjunto de test del entrenamiento* (CTE), que se usa para saber en qué momento se debe detener el entrenamiento. Una red sobreentrenada ha memorizado demasiado, y es incapaz de generalizar a nuevos patrones distintos a los del entrenamiento. O de otra forma, un *sobreentrenamiento* hace que la red ajuste los pesos en exceso, curvando mucho, para ello, la superficie formada por los pesos, de manera que entradas próximas, pero no iguales, obtienen resultados muy diferentes. Mientras que con un entrenamiento menos exhaustivo las entradas próximas darían resultados similares, pues la superficie formada por los pesos es más suave y se produce un proceso de interpolación. El *punto de generalización óptimo* sería aquel a partir del cual la red memoriza en demasía, y antes del cual la red puede mejorar sus resultados con los conjuntos CTE y CE (ver Figura 103).

Una vez terminado el entrenamiento se debe disponer de un tercer conjunto, llamado *conjunto de validación* (CV), con el que se evalúa la red. Es importante que CV se use sólo como comprobación final, pero nunca para entrenar la red. Si no, la red aprenderá este conjunto también, y no se sabrá si es capaz de generalizar.

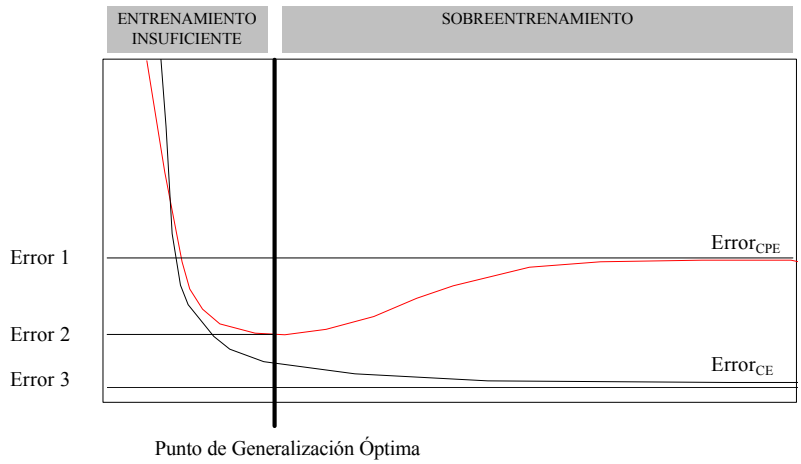


Figura 103.- Si se detiene el entrenamiento antes de alcanzar el punto de generalización óptimo la red aún puede mejorarse. Si se sobreentrena la red pierde la capacidad de generalizar.

A.2. Estructura del perceptrón multicapa

Cumpliendo el esquema que se ha planteado se pueden construir multitud de modelos. Uno de los más sencillos es aquél en que el estado de activación de cualquier célula *j* es función de la entrada total a la misma, siendo la *entrada total* el resultado de sumar todos las salidas correspondientes a las diferentes neuronas que son entradas de *j* multiplicadas cada una por el peso de la conexión que las une y restándole al total una variable que sólo depende de la neurona *j*. Llamaremos a esta variable *umbral* debido a la función inhibidora que puede ejercer. Así la entrada total a la neurona *j* toma la forma de la expresión (A.1).

$$entrada_j = -umbral_j + \sum_i Salida_i \cdot Peso_{i \to j} \qquad (A.1)$$

Para simplificar los cálculos suele cambiarse el umbral por una entrada de peso igual a $umbral_j$, Conectando este axón a una neurona especial que siempre tendrá estado de activación 1. De esta forma la entrada total a la neurona *j* será:

$$entrada_j = \sum_i Salida_i \cdot Peso_{i \to j} \qquad (A.2)$$

De igual forma, las funciones de activación más clásicas son, en el caso discreto, la *función escalón* (A.3), y en el caso continuo, la *función sigmoide* (A.4). Estas funciones hacen que el comportamiento sea tal que si la entrada total es negativa la salida de la neurona sea 0 (ó -1) y que si es positiva la salida sea 1 (Figura 104).

$$salida_j = \begin{cases} 0 \Leftrightarrow Entrada\_Total_j < 0 \\ 1 \Leftrightarrow Entrada\_Total_j \geq 0 \end{cases} \qquad (A.3)$$

$$salida_j = \frac{1}{1 + e^{-Entrada\_Total_j}} \qquad (A.4)$$

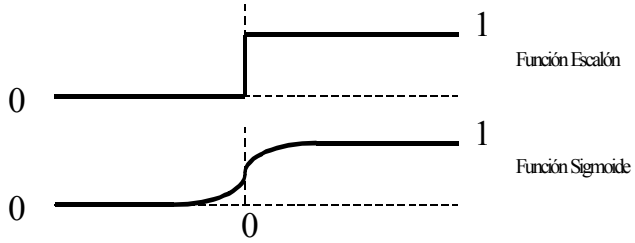


Figura 104.- Aspecto de las funciones de activación escalón (arriba) y sigmoide (abajo).

Así, el modelo mas sencillo consistirá en una red compuesta por una única neurona, en la que la salida viene determinada por la sigmoide (A.4) de la entrada total. Es sencillo demostrar que este esquema es idéntico al del clasificador euclídeo. Así, una red con una sola capa puede separar conjuntos de patrones mediante un hiperplano (ver Figura 105).

Un paso por encima en complejidad se encuentra una red compuesta por varias de estas unidades que operan de forma independiente, es decir, que la salida de cualquiera de ellas no influye en el funcionamiento del resto. Esta configuración, conocida como *perceptrón* de una sola capa, o simplemente perceptrón, equivale a varios clasificadores euclídeos. El *perceptrón multicapa* es la red compuesta por varios niveles sucesivos de perceptrones de una sola capa (ver Figura 106), de manera que las salidas de las unidades de una capa son entradas de las unidades del nivel siguiente. En el siguiente punto se estudiarán las ventajas que aporta está estructura multicapa.

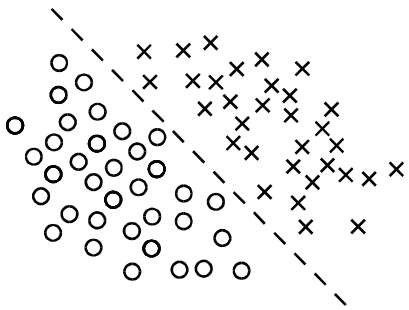


Figura 105.- Patrones 2D separados por un hiperplano (en este caso una recta).

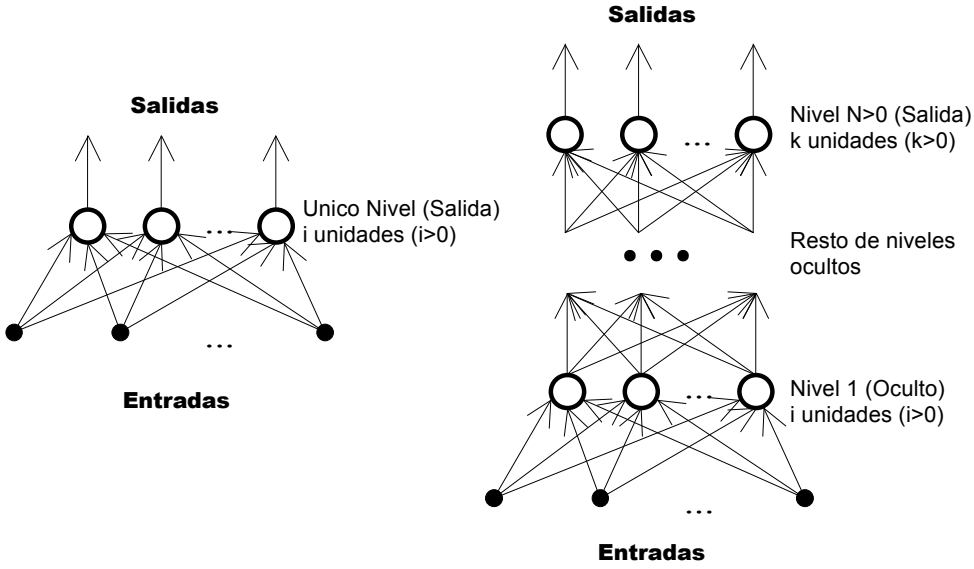


Figura 106.- En estas figuras se presentan los esquemas generales de un perceptrón de simple capa (a la izquierda) y de uno multicapa (a la derecha).

A.3. Propiedades del perceptrón multicapa

El perceptrón multicapa tiene una serie de propiedades que lo hacen especialmente interesante al usarlo como clasificador. Éstas son:

- Robustez frente al ruido aleatorio. Lo que conlleva la selección automática de las características importantes dentro del vector de características que es entrada a la red.

- Separación de regiones complejas dependiendo de la estructura de la red. Una red puede separar clases a pesar de que la hipersuperficie de separación entre ambas sea arbitriamente compleja, o incluso no sea única.

- Capacidad de generalización. Los algoritmos de descenso del gradiente han demostrado en la práctica que obtienen buenas interpolaciones para patrones con los que no se ha entrenado el sistema.

A.3.1 Selección del número de capas ocultas

Se ha dicho que una red con una neurona de salida y sin unidades ocultas, es equivalente a un clasificador euclídeo, siendo capaz de discriminar mediante un hiperplano dos conjuntos de patrones linealmente separables.

Para poder discriminar dos conjuntos sea cual sea la disposición de sus patrones se puede utilizar una red con una sola capa oculta. En los siguientes párrafos se ofrece una demostración informal de este hecho.

Primeramente debe notarse que siempre es posible agrupar los patrones de cualquier clase formando subconjuntos convexos. Estos subconjuntos convexos tienen la particularidad de que pueden separarse linealmente del resto de elementos mediante varios hiperplanos (ver Figura 107). Así, un patrón interior a un subconjunto convexo queda clasificado por la intersección de las regiones definidas por varios hiperplanos.

Es fácil demostrar que este primer paso de separación mediante hiperplanos se puede resolver con una red perceptrón sin unidades ocultas, con una neurona de salida por hiperplano necesario. Cuando, ante un patrón, todas las neuronas de salida de una de estas intersecciones se activan simultáneamente indica que ese patrón pertenece a ese subconjunto convexo.

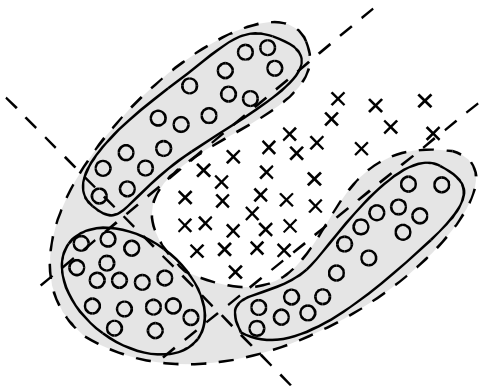


Figura 107.- Los patrones de una muestra siempre se pueden agrupar en regiones convexas.

El siguiente paso consiste en la unión de los resultados de estos subconjuntos convexos en las clases que les dieron origen. Se puede demostrar que esta tarea la puede realizar una red perceptrón, sin unidades ocultas y con una

neurona de salida por cada clase a discriminar. En particular, una neurona de salida, correspondiente a una clase *C*, tendrá un umbral *H*. Para que esta neurona se active sólo cuando un patrón corresponda a la clase *C*, el umbral *H* puede tomarse ligeramente menor al número de hiperplanos necesarios para separar cada uno de los subconjuntos convexos de la clase *C*.

Por tanto, concatenando las dos redes descritas, es decir, construyendo una red con una capa oculta y tantas neuronas de salida como clases, se pueden separar regiones con forma cualquiera.

Así, al menos teóricamente, el comportamiento de cualquier red de dos o más capas se puede obtener con una red de sólo una capa oculta. Sin embargo, los algoritmos de aprendizaje que se conocen no garantizan que esta búsqueda tenga éxito. Es más, la mayoría de las veces estos algoritmos obtienen buenos resultados cuando se disponen varias capas ocultas, y malos resultados cuando sólo se dispone una capa oculta.

A.4. Algoritmos de aprendizaje para el perceptrón multicapa

Resumiendo lo visto, un algoritmo para el ajuste de los pesos de una red de neuronas artificiales debe minimizar alguna función que esté relacionada con el error cometido en las unidades de salida al presentarle una entrada. Si para cada configuración posible de los pesos de una red se conoce el error que comete a la salida al presentarle una entrada determinada, el error puede verse como una superficie cuya altura depende de las coordenadas de los pesos en que se tenga.

La búsqueda del mínimo de estas superficies ha dado lugar a varios algoritmos basados en su mayoría en métodos de descenso del gradiente. Entre todos destaca, por su simplicidad y buenos resultados, el *algoritmo de retropropagación del gradiente*, descubierto en 1974 por Werbos y redescubierto de manera independiente en 1985 por Le Cun y Parker. En la base del algoritmo de retropropagación del gradiente está la *regla delta*.

A.4.1 La regla Delta

Para el perceptrón existe un algoritmo, conocido como regla delta o algoritmo *LMS* ("Least Minimum Squares"), que cambia los pesos, $peso_{ij}$, en la dirección de

máxima pendiente dentro de la superficie del error. Las diferentes componentes del vector de dirección de máxima pendiente vienen determinadas por las derivadas parciales según cada componente de la función error.

$$\Delta peso_{i\to j} = -m \cdot \frac{\P Error}{\P peso_{i\to j}} \tag{A.5}$$

En la ecuación anterior se aprecia que se ha añadido un término m cuya función es determinar la velocidad de variación de los pesos, este término se conoce como *tasa de aprendizaje*. Un valor de m pequeño puede hacer que se caiga fácilmente en mínimos locales, mientras que un valor de m grande puede hacer que se esté siempre oscilando, sin converger nunca. Desgraciadamente el concepto de "grande" y "pequeño" varía según la superficie de error que se esté considerando, la cual a priori es desconocida ya que depende de la muestra que se use para el entrenamiento, por lo que su ajuste constituye un proceso experimental.

Elementos de la red

De (A.5) se deduce que se debe disponer de una medida para el error. La medida más común de éste consiste en la suma de las diferencias cuadráticas entre los valores obtenidos y los deseados, en las *numN* neuronas de salida, después de presentar cada patrón de entrenamiento. Esta fórmula (A.6) se conoce como *error cuadrático medio*.

$$Error = \frac{1}{numN} \sum_{N=1}^{numN} (salida_N - deseada_N)^2 \tag{A.6}$$

Donde se ha tomado la salida de una neurona *j* como una función *F* de la entrada total a la misma (A.2), de acuerdo con la definición de función de activación y función de salida.

$$salida_j = F(entrada_j) \tag{A.7}$$

Cálculo del incremento del peso

Si se aplica sobre (A.6) la regla de la cadena, se puede escribir:

$$\frac{\partial Error}{\partial peso_{i\to j}} = \frac{\partial Error}{\partial entrada_j} \cdot \frac{\partial entrada_j}{\partial peso_{i\to j}} \tag{A.8}$$

De la ecuación (A.2) deducimos:

$$\frac{\partial entrada_j}{\partial peso_{i\to j}} = salida_i \tag{A.9}$$

Definimos ahora:

$$cambio_j = -\frac{\partial Error}{\partial entrada_j} \tag{A.10}$$

Así que uniendo las fórmulas (A.9) y (A.10) se puede reescribir la expresión (A.5), que determina la variación de los pesos en su búsqueda del mínimo, como:

$$\boxed{\Delta peso_{i\to j} = \mu \cdot cambio_j \cdot salida_i} \tag{A.11}$$

El parámetro $cambio_j$ se obtiene aplicando de nuevo la regla de la cadena:

$$cambio_j = -\frac{\partial Error}{\partial entrada_j} = -\frac{\partial Error}{\partial salida_j} \cdot \frac{\partial salida_j}{\partial entrada_j} \tag{A.12}$$

El segundo término de la última fórmula de la ecuación anterior equivale a derivar respecto de $entrada_j$ la ecuación (A.7).

$$\frac{\partial salida_j}{\partial entrada_j} = F'(entrada_j) \tag{A.13}$$

Si la neurona *j* es una neurona de salida, en la ecuación (A.6) uno de los términos considerados será la misma neurona *j*. En tal caso el primer término de la ecuación (A.12) puede calcularse directamente derivando en (A.6) respecto de

$salida_j$. Consecuentemente, la aplicación de esta fórmula para el error se realiza sólo en los casos en que la función de activación de las neuronas es continua y derivable.

$$-\frac{\P Error}{\P salida_j} = (deseada_j - salida_j) \Rightarrow \quad \text{(A.14)}$$

$$cambio_j = (deseada_j - salida_j) \cdot F'(entrada_j) \quad \text{(A.15)}$$

Este algoritmo constituye la regla delta y viene acompañado con un teorema que asegura la convergencia del método siempre que el problema de clasificación tenga solución. Este teorema no asegura, sin embargo, la convergencia hacia el mínimo absoluto de la superficie, sino que el descenso puede ser a un mínimo local de la superficie del error.

Funciones de activación

Como se desprende de (A.15) la aplicación de la regla delta depende de la derivada de la función de activación. La función de activación más común es la sigmoide, pero también suelen usarse la función tangente hiperbólica y la función identidad.

Si la función de activación fuese la identidad se tiene:

$$salida_j = F(entrada_j) = entrada_j \quad \text{(A.16)}$$

Derivando:

$$F'(entrada_j) = 1 \quad \text{(A.17)}$$

Si la función es la sigmoide se puede escribir:

$$salida_j = F(entrada_j) = \frac{1}{1+e^{-entrada_j}} \Rightarrow e^{-entrada_j} = \frac{1}{salida_j} - 1 \quad \text{(A.18)}$$

Derivando:

$$F'(entrada_j) = \left(\frac{1}{1+e^{-entrada_j}}\right)' = \frac{e^{-entrada_j}}{\left(1+e^{-entrada_j}\right)^2} = salida_j \times (1 - salida_j) \quad \text{(A.19)}$$

Según se ha visto la función de activación debe ser derivable, y si no lo es la regla delta se encuentra con un escollo. Éste se debe a la función de error que se está usando. Tomando otro tipo de función para el error se puede seguir aplicando el algoritmo a funciones no derivables.

Por ejemplo, en el caso de la función escalón $F'(entrada_j)$ vale 0 en todo $\hat{A}$ menos en el 0 que no existe. Esto hace que (A.13) no pueda calcularse. La solución es hacer que no sea necesario su cálculo, por ejemplo haciendo que el error tenga tal forma que $cambio_j$ pueda calcularse directamente como la derivada del error respecto de la entrada. Para ello se define m como aquel valor umbral tal que, si la entrada es mayor que él entonces la salida es 1, y si la entrada es menor entonces la salida es 0. Definiendo el error para una neurona determinada con una función derivable respecto de la entrada como (A.20), se puede obtener (A.21) que permite usar la regla delta con la función escalón.

$$Error = \begin{cases} 0 & \text{Si la salida es correcta} \\ \frac{1}{2}(m - entrada_j)^2 & \text{Si la salida es 0 y debería ser 1} \\ -\frac{1}{2}(m + entrada_j)^2 & \text{Si la salida es 1 y debería ser 0} \end{cases} \quad \text{(A.20)}$$

$$cambio_j = -\frac{\P Error}{\P entrada_j} = \begin{cases} 0 & \text{Si la salida es correcta} \\ entrada_j - m & \text{Si la salida es 0 y debería ser 1} \\ entrada_j + m & \text{Si la salida es 1 y debería ser 0} \end{cases} \quad \text{(A.21)}$$

A.4.2 Generalización de la regla Delta

El algoritmo de retropropagación supone una generalización de la regla delta de manera que pueda aplicarse a perceptrones multicapa. En estás redes las fórmulas (A.15) y (A.19) permiten el cálculo del incremento de los pesos para las neuronas

de la última capa. Sin embargo, en las neuronas intermedias no se conoce el valor deseado para su salida, por lo que estas fórmulas no se pueden aplicar directamente.

La idea central de la retropropagación está en que el valor deseado para una neurona de una capa intermedia *J* (ver Figura 108) puede derivarse del valor deseado para las neuronas de la siguiente capa *K*. La aplicación de esta idea de manera recursiva nos conducirá a la última capa para la que sí se conoce el valor deseado a la salida.

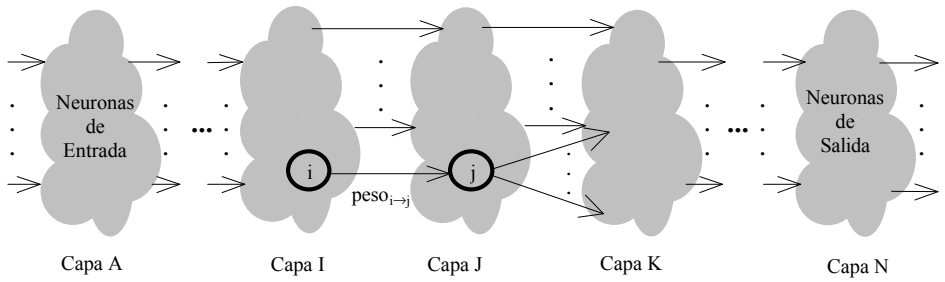


Figura 108.- Esquema general de una conexión entre dos neuronas en un perceptrón multicapa. Se presenta el caso de una conexión cualquiera, que une la neurona i de la capa intermedia I y la neurona j de la capa intermedia J.

En un perceptrón multicapa, el error cometido por la red se puede escribir como una función *E* de las entradas a cualquier capa intermedia de la red, sin más que ir sustituyendo en (A.6) los términos *salida* por sus correspondientes de (A.7). Así para un nivel *J* cualquiera, de un perceptrón multicapa, en el que se encuentran las neuronas 1,2,... *j*,...,*numJ* se puede expresar el error en función de las entradas a este nivel.

$$Error=E(entrada_1, entrada_2,... entrada_j,...,entrada_{numJ}) \tag{A.22}$$

Teniendo en cuenta la definición previa y usando la regla de la cadena se puede simplificar el primer término de la ecuación (A.11) para cualquier neurona de la red.

$$-\frac{\P Error}{\P salida_j} = -\sum_{K=1}^{numK} \frac{\P Error}{\P entrada_K} \cdot \frac{\P entrada_K}{\P salida_j} = \tag{A.23}$$

$$= -\sum_{K=1}^{numK} \frac{\P Error}{\P entrada_K} \cdot \frac{\P \left(\sum_{J=1}^{numJ} peso_{J \to K} \cdot salida_J \right)}{\P salida_j} = \tag{A.24}$$

Realizando la derivada y usando el resultado de la ecuación (A.12) se llega a:

$$= -\sum_{K=1}^{numK} \frac{\P Error}{\P entrada_K} \cdot peso_{j \to K} = \sum_{K=1}^{numK} cambio_K \cdot peso_{j \to K} \tag{A.25}$$

Sustituyendo esta ecuación y la (A.13) en la (A.12) se obtiene:

$$cambio_j = F'(entrada_j) \cdot \sum_{K=1}^{numK} cambio_K \cdot peso_{j \to K} \tag{A.26}$$

Esta fórmula nos permite conocer el factor *cambio* para una capa siempre que se conozca el factor *cambio* para la capa siguiente. Este proceso tiene fin en la última capa de la que se conoce el valor cambio gracias a (A.15). Constituye por tanto un método constructivo para ir calculando el incremento de los pesos que minimiza el error según (A.6), y que exige que la función de activación de cada neurona sea derivable respecto de la entrada total a la misma.

Una vez conocidos estos resultados, se puede resumir el algoritmo de Retropropagación en los pasos que se detallan en el listado adjunto.

- Algoritmo de Retropropagación del Gradiente -

Paso 1.- Iniciar los pesos de las conexiones de la red con valores aleatorios pequeños.

Paso 2.- Presentar uno de los conjuntos de entrada de la muestra a la red y calcular la salida que se obtiene.

Paso 3.- Si no coincide la salida obtenida con la deseada ajustar los pesos como sigue:

$j = N$ (N es el número de neuronas)

$i = j$
Mientras que $j \neq 0$
{
Mientras que $i \neq 0$
{
$D\,peso_{i \to j}(t+1) = m \cdot cambio_j(t) \cdot salida_i(t)$
$peso_{i \to j}(t+1) = peso_{i \to j}(t) + D\,peso_{i \to j}(t+1)$
$i = i - 1$
}
j = j-1
}

En esta forma los pesos se modifican tras cada iteración constituyendo la variante *on-line*. También podrían sumarse cuando han pasado todos los conjuntos de entrada en lo que sería la variante *Batch*.
$cambio_j$ es una medida del error cometido en la neurona j. En este algoritmo este error depende del tipo de neurona. Si la neurona j es de salida, $cambio_j$, tendrá en cuenta la diferencia entre el valor deseado y el obtenido:

$$cambio_j(t) = F'(entrada_j(t)) \cdot (deseado_j(t) - estado_j(t))$$

mientras que si es una neurona intermedia medirá el error de las neuronas a las que alimenta:

$$cambio_j(t) = F'(entrada_j(t)) \cdot S_k(cambio_k(t) \cdot peso_{i \to k}(t))$$

Tomado k los valores de todas las neuronas por encima de la neurona j.

Paso 4.- Si han coincidido todas las salidas obtenidas con las esperadas después de presentar todos las entradas de nuestro conjunto de entrenamiento terminar. En otro caso hacer $t = t + 1$ e ir al paso 2. Como es posible que el proceso no logre obtener siempre la salida deseada frente al patrón de entrada, siempre se puede detener el proceso cuando alcance un valor de error que se considere aceptable.

A.5. Ejemplo de reconocimiento de caracteres a máquina

En este punto se va a presentar un ejemplo real, consistente en el reconocimiento de caracteres numéricos escritos a maquina. Dejando aparte el problema, no trivial, de la segmentación de los caracteres a partir de la imagen de una página escrita a máquina, el problema se puede formular como: tomar una serie de imágenes

correspondientes a caracteres de una fuente concreta (p.e Arial) y determinar a qué carácter corresponde cada una de las imágenes.

A.5.1 Vector de características

En primer lugar se determina el siguiente vector de características discriminantes.

min(ancho/alto,1).-	Un valor entre 0 y 1 que representa las proporciones del carácter si es más ancho que alto.
min(alto/ancho,1).-	Un valor entre 0 y 1 que representa las proporciones del carácter si es más alto que ancho.
densidad normalizada.-	Un valor entre 0 y 1 que mide la proporción de píxeles activos entre píxeles totales de la caja que contiene al carácter.
min(nº objetos / 3,1).-	Un valor entre 0 y 1 proporcional al número de elementos conexos de la imagen si esté numero es menor a tres.
imagen de 8x8.-	La imagen del carácter normalizada a un tamaño de 8x8 y con valores entre 0 y 1.

A.5.2 Construcción de la muestra

El siguiente paso consiste en la construcción de los conjuntos de entrenamiento y de test. Para ello se recopilan 600 caracteres de cada uno de los tipos (ver Figura 109) y se utilizan 360 para entrenamiento, 60 para el proceso de validación durante el entrenamiento y 180 para el test final del clasificador.

Figura 109.- Ejemplo de algunos de los caracteres usados para construir los conjuntos de entrenamiento, validación y test final del clasificador.

A.5.3 Estructura de la red

El vector de características obliga a que la red de neuronas tenga 68 neuronas de entrada (1+1+1+1+64). El número de neuronas de salida viene determinado por el número de clases a reconocer. Así, como se desea reconocer 10 dígitos diferentes, se disponen 10 neuronas de salida.

Por último es necesario decidir si disponer o no capas ocultas, y en caso afirmativo, determinar su número y el número de unidades por capa. En este caso el procedimiento que se sigue consiste en el viejo método de ensayo y error. Comenzando por probar una red sin unidades ocultas.

A.5.4 Entrenamiento y ajuste de la red

Con la muestra construida es posible iniciar el proceso de entrenamiento. Durante el entrenamiento se obtiene un error medio de 0'3 para los patrones de validación. Como el error se considera alto se cambia la estructura de la red y se prueban tres redes con una capa oculta, con 40, 20 y 10 unidades respectivamente. Tras el entrenamiento se obtiene un error medio de 0'05 para la de 40 y la de 20 y de 0'1 para la de 10. En función de estos datos se decide optar por usar la red de 20 unidades ocultas. También se prueba a usar una capa oculta adicional, pero los resultados no ofrecen ninguna mejora.

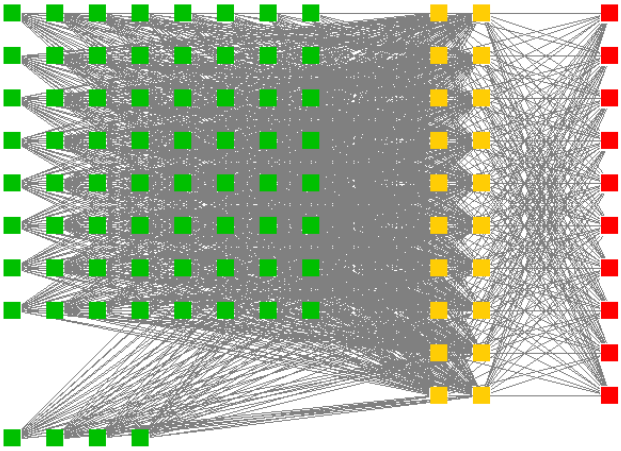


Figura 110.- Imagen de una red de neuronas con 68 unidades de entrada (verde), 20 unidades ocultas (amarillo) y 10 de salida (rojo).

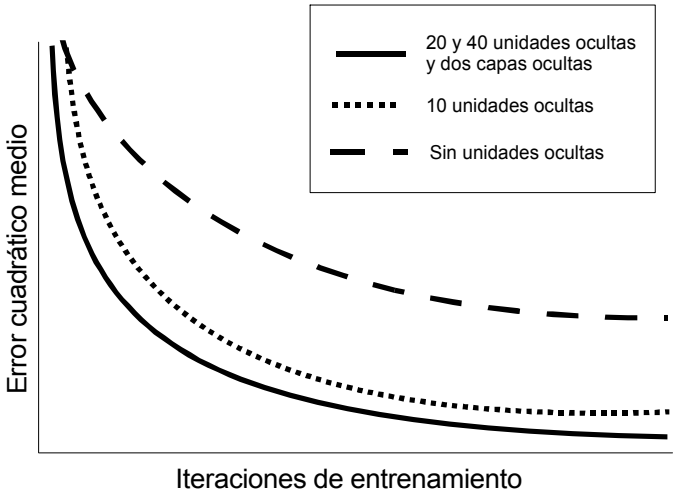


Figura 111.- Resultado obtenido durante el entrenamiento de varios perceptrones multicapa con diferentes estructuras, para el problema de reconocimiento de dígitos.

Tras elegir la configuración de red se vuelve a entrenar utilizando el conjunto de entrenamiento y el de validación del entrenamiento para determinar el punto óptimo de generalización en el que se debe detener el entrenamiento.

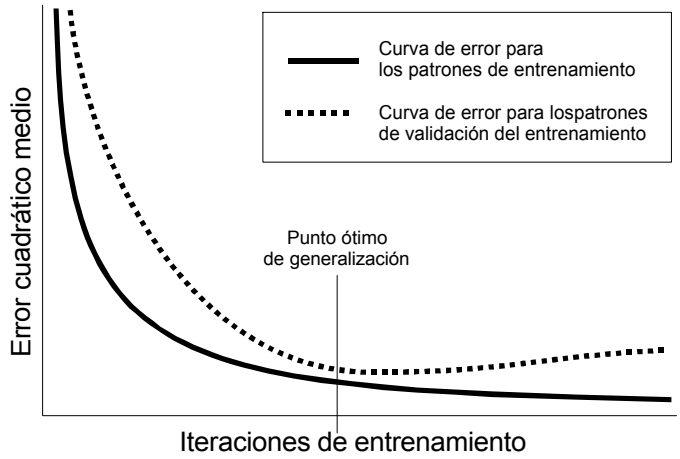


Figura 112.- Curva que presenta el error obtenido con los patrones de entrenamiento y con los de validación del entrenamiento. El punto óptimo de generalización indica el punto en el que se debe interrumpir el entrenamiento.

		Patrón leído por la red									
		0	1	2	3	4	5	6	7	8	9
Patrón presentado	0	16								2	
	1		17						1		
	2			17					1		
	3				17					1	
	4					18					
	5						17	1			
	6						1	17			
	7			1					17		
	8									18	
	9										18

Tabla 7

Los resultados obtenidos al presentar el conjunto de test aparecen en la Tabla 7. Esta tabla se conoce como *matriz de confusión* y presenta cada reconocimiento mediante un par de coordenadas, la horizontal indica que elemento se presento al clasificador, y la vertical en que clase lo ha clasificado. Se observa que los elementos que se encuentran exactamente sobre la diagonal noroeste de la matriz corresponden a los patrones que han sido identificados con éxito.

A.5.5 Bibliografía del anexo

[HKP91] cap. 6,

[Loo97] cap. 5

[Z+95]

Anexo B Referencias Bibliográficas

B.1 Bibliografía básica

[Bax94] G.A. Baxes, *Digital Image Processing: Principles and Applications*, J. Wiley & Sons, 1994.

[Esc01] A. de la Escalera, *Visión por computador: Fundamentos y métodos*, Pearson- Prentice Hall, 2001.

[F+97] J.D. Foley, A. van Dam, S.K. Feiner y J.F. Hughes, *Computer Graphics: Principles and Practice*, 2nd edition in C, Addison-Wesley, 1997.

[Gon00] J. González **,** *Visión por computador*, Ed. Paraninfo, 2000.

[GW93] R.C. González y R.E. Woods, *Digital Image Processing*, Addison Wesley, 1993.

[HKP91] J. Hertz, A. Krogh y R.G. Palmer, *Introduction to the Theory of Neural Computation*, Addison Wesley, 1991.

[JKS95] R. Jain, R. Kasturi y B.G. Schunk, *Machine Vision*, McGraw-Hill, 1995.

[Loo97] C.G. Looney, *Pattern Recognition using Neural Networks: Theory and Algorithms for Engineers and Scientists*, Oxford University Press, 1997.

[Mar93] D. Maravall, *Reconocimiento de Formas y Visión Artificial*, Ed. Ra-Ma, 1993.

[Par97] J.R. Parker, *Algorithms for Image Processing and Computer Vision*, J. Wiley and Sons, 1997.

[SHB99] M. Sonka, V. Hlavac y R. Boyle, *Image Processing, Analysis and Machine Vision,* PWS Publishing, 1999.

[TV98] E. Trucco y A. Verri, *Introductory Techniques for 3-D Computer Vision*, Prentice Hall, 1998.

[Z+95] A. Zell et al, Stuttgart Neural Network Simulator: User Manual v. 4.0, 1995.

B.2 Bibliografía adicional

La bibliografía adicional está formada por otras referencias que permitirán completar el estudio de los temas explicados en esta obra.

[BS96] K. Bowyer y G. Stockman (organiz.), "Themes for Improved Teaching of Image-Related Computation", *1997 IEEE Computer Society Workshop on Undergraduate Education and Image-Related Computation*, disponible en la web: http://marathon.csee.usf.edu/teaching_resources.html, 1997.

[B+00] K. Bowyer et al (eds.), Proc. Second IEEE CS Workshop on Undergraduate Education & Image Computation, disponible en: http://figment.csee.usf.edu/educ-ws-00.html, 2000.

[BB88] G. Brassard y P. Bratley, *Algorithmics. Theory and Practice*, Prentice-Hall, 1988.

[Cas96] K. Castleman, *Digital Image Processing*, 2nd Edition, Prentice-Hall, 1996.

[CC01] ACM/IEEE Curriculum Committee on Computer Science, "ACM/IEEE-CS Computing Curricula 2001", *CC2001 Steelman Draft*, 2001. Disponible en la web: http://www.acm.org/sigcse/cc2001/steelman.

[CI01] D. Cojocaru y M. Ivanescu, "An Analyse of a Computer Vision Course for EE Students", *Proc. IASTED Intl. Conf. On Visualization Imaging, and Image Processing*, sept. 2001, pp. 636-641.

[Dom94] A. Domingo Ajenjo, *Tratamiento digital de imágenes*, Ed. Anaya Multimedia, 1994.

[Ett97] D.M. Etter, *Solución de Problemas de Ingeniería con Matlab*, 2ª Edición, Prentice Hall, 1997.

[Fau93] O.D. Faugeras, *Three-Dimensional Computer Vision: A Geometric Viewpoint*, MIT Press, 1993.

[Fau00] M. Faúndez Zanuy, *Tratamiento Digital de Voz e Imagen y Aplicación a la Multimedia*, Marcombo, 2000.

[FH01] E. Fink y M. Heath, "Image-Processing Projects for an Algorithms Course", *International Journal on Pattern Recognition and Artificial Intelligence*, vol. 15, nº. 5, agosto 2001, pp. 859-868.

[GH01] A. Gruen y T.S. Huang, *Calibration and Orientation of Cameras in Computer Vision*, Springer, 2001.

[Gue99] C. Guerra, "Visión and Image Processing Algorithms", en: *Algorithms and Theory of Computation Handbook* (M.K. Atallah, ed.), CRC Press, 1999.

[HS90] E. Horowitz y S. Sahni, *Fundamentals of Data Structures in Pascal*, Computer Science Press, 3ª edición, 1990.

[HS92] R. Haralick and L. Shapiro, *Computer and Robot Vision*, Addison-Wesley, 1992.

[JHG99] B. Jahne, H. Hauecker and P. GeiBler (editores) , *Computer Vision and Applications Handbook* (vol. III: *Systems and Applications*), Academic Press, 1999.

[Kab99] I. Kabir, *High Performance Computer Imaging*, Manning, 1999.

[KSK98] R. Klette, K. Schlüns y A. Koschan, *Computer Vision: Three-Dimensional Data From Images*, Springer, 1998.

[Mat99] Matrox Electronic Systems Ltd, *Matrox Imaging Library: User Guide v. 6.0*, 1999.

[Mat01] The Mathworks, *Matlab Image Processing Toolbox v 3.1: User's Gu*ide, 2001.

[Max98] B. A. Maxwell, "Teaching computer vision to computer scientists: issues and a comparative textbook review", *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 12, nº 8, pp. 1035-1051, agosto 1998.

[Max01] B. A. Maxwell, "A Survey of Computer Vision Education and Text Resources", *International Journal on Pattern Recognition and Artificial Intelligence*, vol. 15, nº. 5, pp. 757-774, agosto 2001.

[Mic96] Microsoft Corportation, *Microsoft Win32 Programmer's Reference*, 1996.

[Mur98] R.R. Murphy, Teaching Image Computation in an Upper Level Elective on Robotics, *International Journal on Pattern Recognition and Artificial Intelligence*, vol. 12, nº 8, 1998.

[MvR94] J.D. Murray y W. Van Ryper, *Encyclopedia of Graphics File Formats*, 2nd edition, o´Reilly & Associates, 1994.

[MW93] H.R. Myler y A.R. Weeks, *Computer Imaging Recipes in C*, Prentice Hall, 1993.

[PC01] G. Pajares y J.M. de la Cruz, *Visión por computador: Imágenes digitales y aplicaciones*, Ed. Ra-Ma, 2001.

[PG01] M.W. Powell y D. Goldgof, "Software Toolkit for Teaching Image Processing", *International Journal on Pattern Recognition and Artificial Intelligence*, vol. 15, nº. 5, agosto 2001, pp. 833-844.

[SV+01] A. Sánchez, José F. Vélez, A.B. Moreno y J.L. Esteban, "Introducing Algorithm Design Techniques in Undergraduate Digital Image Processing Courses", *International Journal on Pattern Recognition and Artificial Intelligence*, vol. 15, nº. 5, pp. 789-803, agosto 2001.

[SG98] S. Sarkar y D. Goldgof, Integrating Image Computation in Undergraduated Level Data-Structure Education, *International Journal on Pattern Recognition and Artificial Intelligence*, vol. 12, nº 8, 1998.

[SMB98] J.L. Starck, F. Murtagh y A. Bijaoui, *Image Processing and Data Analysis. The Multiscale Approach*, Cambridge University Press, 1998.

[SS01] G. Stockman y L. Shapiro, *Computer Vision*, Prentice-Hall, 2001.

[Tsa87] R. Y. Tsai, "A Versatile Camera Calibration Technique for High Accuracy 3D Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses", *IEEE Trans. on Robotics and Automation*, Vol RA-3 (4), pp. 323-344, 1987.

[Ull96] S. Ullman, *High-level Vision: Object Recognition and Visual Cognition*, MIT Press, 1996.

[Umb99] S.E. Umbaugh, *Computer Vision and Image Processing: A Practical Approach using CVIPtools*, Prentice Hall, 1999.

[Ver91] D. Vernon, *Machine Vision: Automated Visual Inspection and Robot Vision, Prentice Hall*, 1991.

[Zue00] N. Zuech, *Understanding and Applying Machine Vision*, Marcel Dekker, 2000.

B.3 Material complementario

Aparte de los libros, existen otras fuentes de información relacionadas con los temas abordados en la asignatura (principalmente artículos de revistas

especializadas y páginas Web), que podrían complementar a los contenidos explicados en clase, dar a conocer el estado de la investigación actual en estos temas y ofrecer una ayuda inestimable para la realización de las prácticas. A continuación, se reseñan algunas de estas fuentes de información.

B.3.1 Revistas

El lector interesado en el tratamiento digital de imágenes y visión artificial puede consultar multitud de revistas dedicadas a la materia explicada. Tienen especial relevancia:

- *Computer Vision and Image Understanding* (http://www.apnet.com/www/journal/iv.htm).
- *IEEE Trans. Pattern Analysis and Machine Intelligence* (http://www.computer.org/tpami/).
- *Pattern Recognition* (http://www.elsevier.nl/inca/publications/store/3/2/8/).
- *IEEE Transactions on Image Processing* (http://www.ieee.org/organizations/pubs/pub_preview/ip_toc.html).
- *Pattern Recognition Letters* (http://www.elsevier.nl/inca/publications/store/5/0/5/6/1/9/).
- *Image and Vision Computing* (http://www.elsevier.nl/locate/imavis).
- *IJCV - International Journal of Computer Vision* (http://kapis1.wkap.nl/kapis/CGI-BIN/WORLD/journalhome.htm?0920-5691).
- *Journal of Mathematical Imaging and Vision* (http://www.ics.forth.gr/ecvnet/publications/jmiv/jmiv.html).
- *Real-Time Imaging* (http://www.academicpress.com/rti).
- *International Journal on Pattern Recognition and Artificial Intelligence* (http://www.worldscinet.com/journals/ijprai/ijprai.shtml).
- *IEEE Transactions on Medical Imaging* (http://www.ieee.org/organizations/pubs/pub_preview/mi_toc.html).
- *Machine Vision and Applications* (http://link.springer.de/link/service/journals/00138/).
- *IEE Proceedings - Vision, Image and Signal Processing* (http://www.iee.org.uk/Publish/Journals/Profjourn/Proc/vis/).
- *Advanced Imaging* (http://www.advancedimaging.com).

Entre las principales editoriales de publicaciones (libros, revistas u material en cualquier otro soporte), relacionados con estas materias, hay que mencionar las siguientes:

- *IEEE Press.*
- *Springer Verlag.*
- *Academic Press.*
- *Prentice Hall.*
- *Kluwer Academic Publishers.*
- *Elsevier Science.*
- *Addison Wesley.*
- *World Scientific Publishing Company.*
- *John Wiley & Sons.*
- *MIT Press.*
- *CRC Press.*

B.3.2 Software

Existen numerosas heramientas *software* que pueden adaptarse, total o parcialmente para la enseñanza de la visión por computador. Las mejores herramientas son de pago, aunque también hay *software* libre que se puede utilizar para estos propósitos. En la página *web: The Computer Vision Homepage* de la Universidad de Carnegie Mellon (cuyo enlace aparece en esta sección) existe un apartado de *software* donde puede encontrarse una lista extensa de programas, librerías, entornos de programación software para imágenes. Remitimos al lector a la consulta de dicha página, donde también aparecen los enlaces *web* de las herramientas referenciadas. Se referencian, a continuación algunas más conocidas:

- *Image Processing Toolbox de MATLAB.*
- *Khoros.*
- *Matrox Imaging Library (MIL).*
- *Tina.*
- *VISILOG.*
- *HIPS .*

B.3.3 Imágenes de test

Cuando se trata de demostrar el resultado de una determinada técnica (novedosa o no) de visión artificial, los investigadores y demás usuarios se han puesto de acuerdo en utilizar un conjunto de imágenes de test comunes. Por ejemplo, son

conocidas las imágenes de: Lena, el hombre de la cámara, etc. También para determinadas aplicaciones se han creado bases de datos conteniendo una muestra de imágenes lo suficientemente grande, que libra al usuario en muchos casos de crearla. Para una información más detallada sobre imágenes de prueba en aplicaciones de visión, se remite (nuevamente al lector) a la página: ***:*** *The Computer Vision Homepag,* y dentro de ella al enlace:

http://www-2.cs.cmu.edu/afs/cs/project/cil/ftp/html/v-images.html, donde se pueden encontrar información de muchas bases de datos, imágenes asiladas y secuencias de imágenes estándar de prueba.

B.3.4 Paginas Web

Las páginas web relacionadas con esta materia han sido subdivididas en dos grupos: aquéllas que son más generales y las relacionadas con la enseñanza.

Páginas web sobre tratamiento digital de imágenes y visión computacional

· *The Computer Vision Homepage at Carnegie Mellon University*
(http://www-2.cs.cmu.edu/afs/cs/project/cil/ftp/html/vision.html)

Creada en la Universidad de Carnegie Mellon University en 1994, es la página más conocida sobre tratamiento digital de imágenes y visión computacional. Centraliza una colección de enlaces *web* orientados hacia la investigación en tratamiento de imágenes y visión por computador. La página está organizada en subpáginas que contienen, respectivamente, la siguiente información:

- Grupos de investigación que trabajan en visión artificial.
- Hardware: sistemas de investigación y productos comerciales
- Software: código de programas para investigación, librerías para procesamiento de imágenes, generadores de datos sistéticos, etc.
- Imágenes de test.
- Congresos sobre visión artificial.
- Publicaciones: referencias de artículos en congresos y revistas, libros, tutoriales, etc.
- Información general: grupos de noticias, FAQs, archivos, etc.

Otros enlaces relacionados

· *CVonline: The Evolving, Distributed, Non-Proprietary, On-Line Compendium of Computer Vision*
(http://www.dai.ed.ac.uk/CVonline/CVentry.htm).

Se trata de una colección de resúmenes en hipertexto sobre las aspectos principales de la visión artificial. Está página está mantenida por Robert Fisher de la Universidad de Edimburgo. El índice está organizado en alrededor de 700 puntos. La idea ofrecer a la comunidad científica que trabaja en estos temas una colección de material común de manera gratuita. El compendio trata de ser un resumen de métodos y aplicaciones de la visión por computador, estando organizado por secciones que abarcan las áreas más importantes de investigación y de aplicación práctica. A continuación, se enumeran algunos de los tópicos que aparecen al primer nivel de la jerarquía de hiperenlaces:

1. Aplicaciones.
2. Bases de datos e índices.
3. Sistemas de visión famosos.
4. Técnicas de visión genéricas.
5. Métodos de extracción de características geométricas.
6. Física de la imagen.
7. Transformaciones y filtrados sobre imágenes.
8. Movimiento, seguimiento y análisis de secuencias de imágenes.
9. Hardware para tratamiento de imágenes.
10. Modelos de representación de objetos, del mundo y de escenas.
11. Métodos de reconocimiento y de registro.
12. Interpretación de imágenes.

· *USC Annotated Computer Vision Bibliography*
(http://iris.usc.edu/Vision-Notes/bibliography/contents.html).

Es una colección de bibliografía comentada, mantenida en la sobre visión por ordenador, tratamiento de imágenes y otras áreas relacionadas. Contiene una introducción que describe todo lo que está contenido en esta página, como referenciar la bibliografía y tópicos relacionados. He aquí la estructura (incompleta) del índice principal:

1. Ayuda, FAQs, Cómo localizar entradas y conseguir artículos

2. Nombres de revistas y listado de conferencias
3. Libros y *surveys*
4. Técnicas de bajo nivel, sensores, representación de imágenes, ...
5. Tratamiento del color, codificación y restauración de imágenes.
6. Detección y análisis de bordes y líneas .
7. Análisis de características 2D y textura .
8. Técnicas de segmentación de regiones 2D.
9. Extracción de formas 3D.
10. Visión estereoscópica.

· *Machine Vision Online*
(http://www.machinevisiononline.org).

Se trata de una sociedad americana dedicada a la visión por computador desde una perspectiva más industrial y profesional. Ofrece en sus enlaces una guía de compradores, consejos para aplicar con éxito la visión por computador, recursos educativos, eventos, enlaces, etc.

Páginas Web sobre cursos y sobre la enseñanza de la visión computacional

· *Computer Vision Courses*
(http://www.palantir.swarthmore.edu/~maxwell/visionCourses.htm).

Página creada por Bruce A. Maxwell, Swarthmore College en el 2000. Contiene enlaces a 33 cursos diferentes sobre tratamiento de imágenes y visión computacional impartidos en distintas universidades del mundo (en esa página aparece una referencia y un enlace a nuestro curso en la Universidad Rey Juan Carlos). Algunos de los cursos presentados tienen un enfoque clásico; otros presentan una orientación interdisciplinar (por ejemplo, cursos que ralacionan la visión con la percepción biológica, los interfaces de usuario, los gráficos por computador, etc). Muchos de los cursos incluyen su propio material docente usado: apuntes del curso, prácticas de laboratorio, enlaces de interés, etc.

· *Hypermedia Image Processing Reference*
(http://www.dai.ed.ac.uk/HIPR2).

Hypermedia Image Processing Reference(HIPR) es una fuente de asistencia on-line para los usuarios de sistemas basados en el tratamiento de imágenes en todo el

mundo. *HIPR* ha sido desarrollada en el Departamento de Inteligencia Artificial de la Universidad de Edimburgo con vistas a proporcionar material de tipo tutorial para cursos sobre tratamiento de imágenes y visión por computador. El paquete software desarrollado ofrece información *on-line* sobre un gran número de operaciones de tratamiento de imágenes, con numerosos ejemplos sobre imágenes digitalizadas. Entre los servicios ofrecidos por *HIPR* están:

- Referencias sobre unas cincuenta clases de operaciones sobre imágenes más habituales.
- Descripción detallada sobre cómo trabaja cada operación.
- Una demostración en JAVA sobre cada operación y el código fuente de los algoritmos.
- Ejemplos de imágenes de datos y de resultados para mostrar gráficamente cómo funciona cada operación.
- Un gran número de ejercicios.
- Información bibliográfica.
- Tablas de operadores equivalentes en varios paquetes de tratamiento de imágenes: VISILOG, Khoros, la librería de tratamiento de imágenes (toolbox) de MATLAB y HIPS.

B.3.5 Asociaciones relacionadas con visión computacional

· *IEEE Computer Society*
(www.computer.org).
La *IEEE Computer Society* es una de las 36 sociedades dentro de IEEE (*Institute of Electrical and Electronic Engineers*). Hoy día, IEEE es la sociedad profesional técnica mayor del mundo (370.000 asociados en 150 paises). *IEEE Computer Society*, fundada en 1946, tiene 100.000 asociados y su objetivo es proporcionar información técnica y servicios a los profesionales de la informática a nivel mundial. Por medio de las conferencias organizadas, publicaciones, capítulos de estudiantes, comités técnicos, y grupos de trabajo, se promueve el intercambio de información y la innovación tecnológica entre sus miembros.

En relación con las conferencias promovidas por *IEEE Computer Society* sobre visión artificial conviene destacar: la *Conference on Computer Vision and Pattern Recognition* (CVPR) y la *International Conference on Computer Vision* (ICCV).

· *International Association for Pattern Recognition (IAPR)*
(http://www.iapr.org/ind2.html).
La Asociación Internacional sobre Reconocimiento de Patrones (IAPR) es una asociación científico-profesional no lucrativa involucrada en temas de reconocimiento de patrones, visión artificial y tratamiento de imágenes en un sentido amplio. Esta organizada por paises (una organización por país) y las personas interesadas en participar en las actividades organizadas por la IAPR realizadan dichas actividades adhiriéndose a su organización nacional. IAPR nace en 1978 y organiza bienalmente el Congreso Internacional sobre Reconocimiento de Patrones. También, co-esponsoriza algunas otras conferencias en esta área. Para conocer en detalle los objetivos y actividades de esta organización, se remite al lector a la página *web* de la asociación.

· *Asociación Española de Reconocimiento de Fromas y Análisis de Imágenes (AERFAI)*
(http://decsai.ugr.es/aerfai/information.html).
Se trata de la asociación española que desde 1982 constituye la organización española dentro de la IAPR. La AERFAI también organiza cada dos años un congreso nacional: el Simposio Nacional de Reconocimiento de Formas y Análisis de Imágenes. Además, publica periódicamente un boletín para los miembros de la asociación y algunos libros sobre temas relacionados a nivel nacional.

Anexo C

Índice alfabético

N

O

P

V

W