

# 3. PROCESAMIENTO DE IMÁGENES

Este capítulo comienza con una visión general de la teoría de procesamiento de imágenes, para centrarse a continuación en los diferentes tipos de filtros y métodos de análisis existentes en la actualidad. Se estudiarán diferentes tipos de transformaciones, y finalmente se hará una breve descripción de los sistemas de vídeo digital, dejando el procesamiento de vídeo en FPGA para el siguiente capítulo.

## 3.1 Imágenes digitales

### 3.1.1 Definición de una imagen digital

Una imagen se define como una función de dos dimensiones  $f(x,y)$  donde  $x$  e  $y$  son las coordenadas de un plano que contiene todos los puntos de la misma, y  $f(x,y)$  es la amplitud en el punto  $(x,y)$  a la cual se le llama intensidad o nivel de gris de la imagen en ese punto. En el caso de que tanto las coordenadas  $x$  e  $y$  como los valores de intensidad de la función  $f$  sean discretos y finitos, se habla de una imagen digital [41].

Una imagen digital está compuesta de un número finito de elementos y cada uno tiene una localidad y un valor particular. A estos elementos se les llama puntos elementales de la imagen o píxeles, siendo este último el término comúnmente utilizado para denotar la unidad mínima de medida de una imagen digital.

En la **Figura 3.1** se muestra una representación de una imagen con 256 niveles de intensidad. En ella, cada uno de los píxeles está representado por un número entero que es interpretado como el nivel de intensidad luminosa en la escala de grises. Ampliando la imagen en una zona cualquiera, se pueden apreciar estos valores, que se muestran en forma de matriz en la misma figura, correspondiéndose cada elemento de la matriz  $N_{ij}$  con las coordenadas en el plano  $x=i, y=j$ .

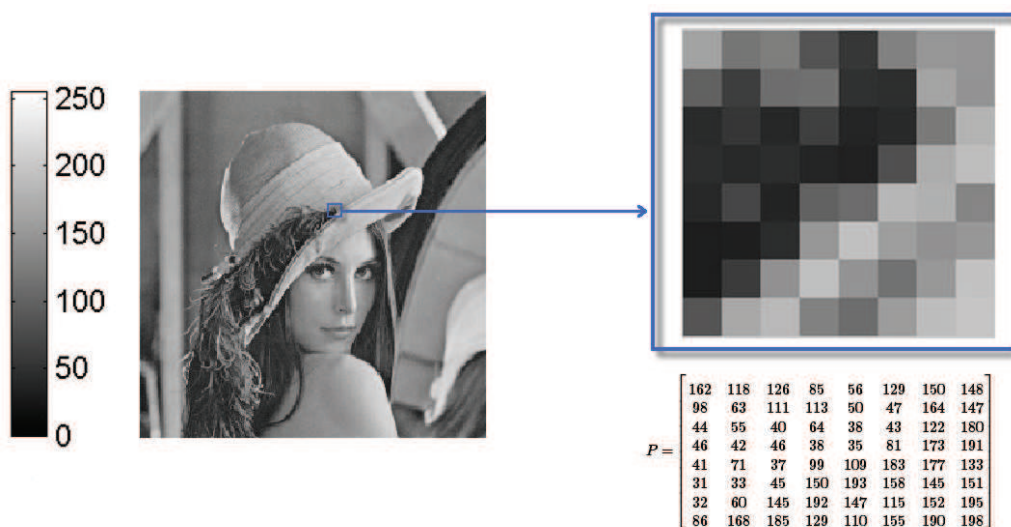


Figura 3.1. Imagen con 256 niveles de intensidad y representación numérica de un fragmento 8x8.

Uno de los parámetros de mayor importancia en una imagen digital es su resolución. La resolución es la cantidad de píxeles que contiene una imagen. Se utiliza también para clasificar casi todos los dispositivos relacionados con las imágenes digitales, ya sean pantallas de ordenador o televisión, impresoras, escáneres, cámaras digitales, etc. La resolución total expresa el número de píxeles que forman una imagen de mapa de bits. La calidad de una imagen depende directamente de su resolución. Es común expresar la resolución de una imagen en dos valores numéricos, donde el primero es la cantidad de columnas de píxeles (cuántos píxeles tiene la imagen de ancho) y el segundo es la cantidad de filas de píxeles (cuántos píxeles tiene la imagen de alto).

### 3.1.2 Imágenes en color

El fundamento para describir una imagen digital en color es el mismo que el expuesto anteriormente, con la salvedad de que cada elemento o píxel es descrito y codificado de otra forma, según el espacio de color que se esté utilizando. Así por ejemplo, para un espacio de color RGB (generalmente el más usado para representar imágenes), se representa cada píxel como un color creado a partir de ciertas cantidades de los colores rojo, verde y azul [42]. Esta representación se puede interpretar como una matriz de tres niveles de intensidad, donde cada nivel corresponde a la intensidad de color de las componentes rojo, verde y azul, como se muestra en la **Figura 3.2** y **3.3**.

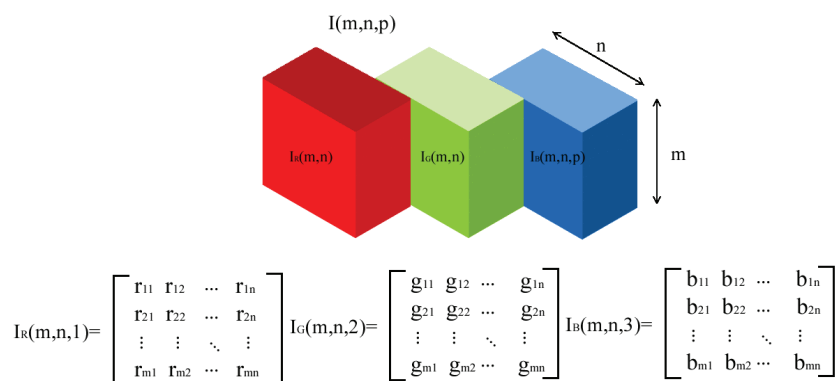


Figura 3.2. Planos de color RGB representados como tres matrices bidimensionales.

Trabajar con matrices para describir imágenes es habitual en el campo de la visión por computador. Sin embargo, como se verá más adelante, ésta no es la única forma de representar una imagen en color, y cuando se trata de procesar imágenes en FPGA, se pierde el concepto de matriz para dar lugar al concepto de flujo de datos por bus.

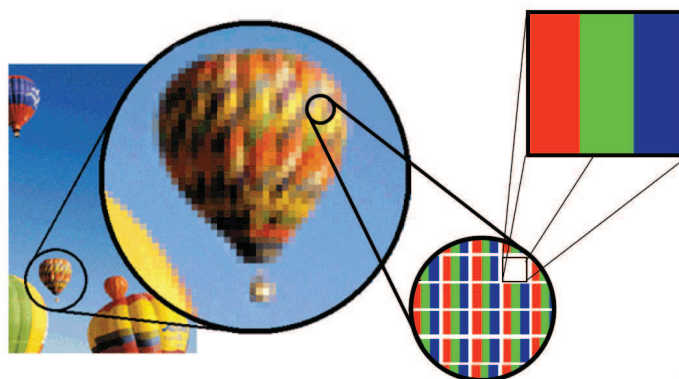


Figura 3.3. Componentes primarias en los píxeles de una imagen en color [61].

En una imagen RGB, cada píxel está compuesto por un valor de intensidad correspondiente a cada componente primaria. El color resultante del píxel vendrá por tanto definido por la "cantidad" de intensidad que tenga cada componente. Así, el color blanco estará compuesto de la máxima intensidad de color para los tres componentes. Por el contrario, el color negro será el resultado de reducir al mínimo la intensidad de los componentes (**Figura 3.4**).

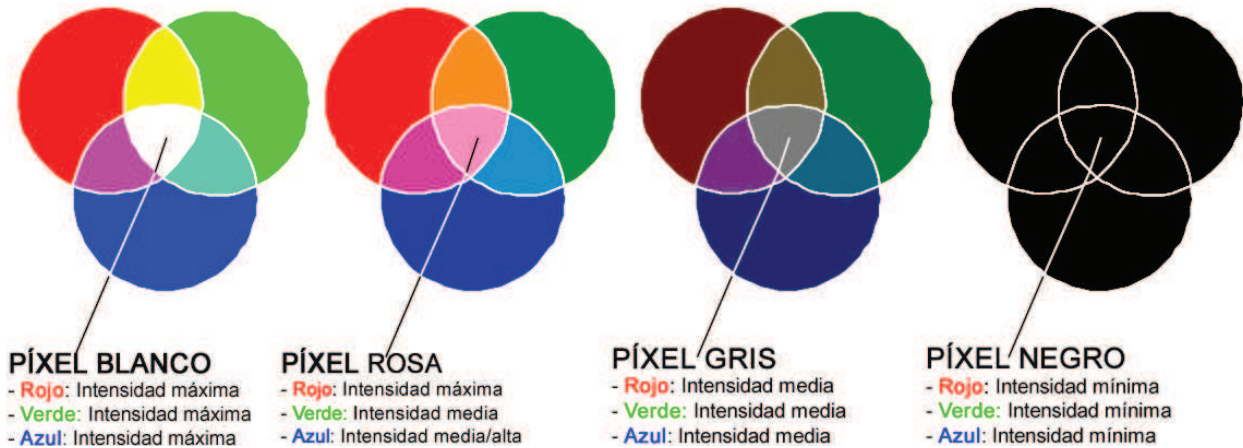


Figura 3.4. Píxel resultante a partir de la intensidad de sus componentes (Imagen modificada sobre [61]).

### 3.1.3 Variables del color

Para comprender mejor por qué existen diferentes espacios de color, se comenzará haciendo una breve descripción de las propiedades innatas del color, que son el matiz, la luminosidad, el tono y la saturación.

**El matiz (Hue).** Es el valor cromático de un color, la frecuencia del espectro donde se encuentra. Depende de la longitud de onda dominante, y es la cualidad que permite clasificar a los colores como amarillo, rojo, violeta, etc.

**La luminosidad (Lightness).** Es el resultado de la mezcla de los colores con blanco o negro y tiene referencia de matiz. Representa la cantidad de luz presente en un color, más blanco o más negro, según sea el caso. Cuanto mayor es la luminosidad, mayor es la cantidad de luz en un color, es decir, más color blanco posee.

**La saturación (Saturation).** Se refiere al grado de pureza de un color y se mide con relación al gris. Los colores con menor saturación se muestran más agrisados, con mayor cantidad de impurezas y con menor intensidad luminosa.

La saturación es uno de los principales retos en el campo del procesamiento de imágenes, debido a que los colores son cada vez más difíciles de distinguir unos de otros a medida que su saturación disminuye, teniendo que usar diferentes espacios de color para optimizar los algoritmos de detección. Esta situación es especialmente delicada en los procesos de segmentación, como se verá en posteriores capítulos.

## Saturación



Figura 3.5. Escala de saturación de las componentes RGB.

### 3.1.4 Espacios de color

Los Espacios de Color son una herramienta importante en el procesamiento digital de imágenes, ya que permiten analizar cada píxel desde otro punto de vista, y así aprovechar toda la información presente dentro de la imagen. Los trabajos más recientes realizados en este área se relacionan con la segmentación de imágenes a color, la localización de objetos, análisis de textura, morfología matemática, estandarización de imágenes a color, entre otros. Los sistemas no lineales son frecuentes en los Espacios de Color, ya que éstos buscan realzar ciertas particularidades de una imagen [43].

Existen numerosos Espacios de Color, atendiendo cada uno a necesidades tan dispares que van desde la fisiología del ojo humano (Espacio de Hering, o espacio de colores oponentes), hasta el modelo de color sustractivo usado en la impresión sobre papel (CMYK). En este apartado se presentan los más utilizados en la visión por computador y en el procesamiento de imágenes. Cabe destacar que algunos de estos Espacios de Color no tienen como objetivo hacer la visualización de colores más fiel a la realidad, sino que son abstracciones matemáticas, generalmente no lineales, que hacen posible el tratamiento de ciertas propiedades de la imagen.

#### 3.1.4.1 El modelo RGB

El modelo RGB es uno de los más utilizados por los sistemas informáticos para crear y reproducir los colores en monitores y pantallas. Está basado en la llamada "síntesis aditiva", donde las intensidades de luz relativas al rojo, al verde y al azul son sumadas entre sí para conseguir los distintos colores, incluyendo el negro y el blanco.

La representación gráfica del modelo RGB (**Figura 3.6**) se realiza mediante un cubo unitario con los ejes R, G y B. El origen (0,0,0) representa el negro y las coordenadas (1,1,1) el blanco. Los vértices del cubo en cada eje R, G y B, de coordenadas (1,0,0), (0,1,0) y (0,0,1) representan los colores primarios rojo, verde y azul. Los restantes tres vértices (1,0,1), (0,1,1) y (1,1,0) al magenta, cian y amarillo respectivamente, colores secundarios y respectivamente complementarios del verde, rojo y azul. La diagonal del cubo representa la gama de grises desde el negro al blanco. En esta diagonal cada punto o color se caracteriza por tener la misma cantidad de cada color primario.

Las imágenes con modelo RGB contienen tres planos de imágenes independientes, uno para cada color primario. Cuando estas tres imágenes son proyectadas a un monitor RGB, la pantalla de fósforo produce una imagen de color compuesto. El procesamiento de imágenes en color, utilizando el modelo RGB, toma sentido cuando las imágenes se expresan naturalmente en términos de tres planos de color.

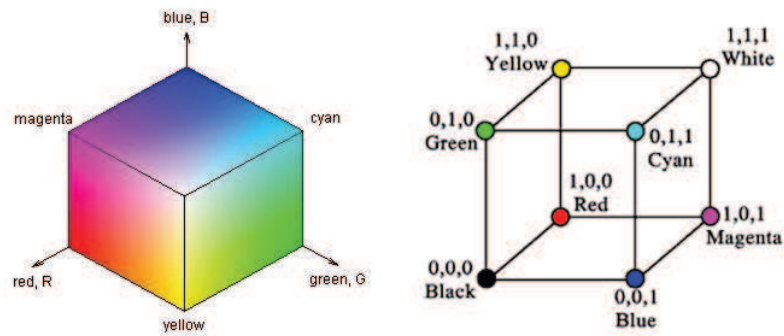


Figura 3.6. Representación gráfica del modelo RGB [168]

Lo anterior convierte al modelo RGB en un modelo de gran importancia para el procesamiento de imágenes, a pesar de que no deriva en un proceso intuitivo para determinadas aplicaciones como por ejemplo la de comparar colores.

### 3.1.4.2 El modelo HSV

Las siglas H, S y V corresponden a Tono (hue), Saturación (saturation) y valor (value) respectivamente. También se denomina HSB, siendo B el brillo (brightness). El sistema coordinado es cilíndrico, y el subconjunto de este espacio donde se define el color es una pirámide de base hexagonal (Figura 3.7) [44].

En el modelo HSV los colores más brillantes están contenidos en el área hexagonal correspondiente a  $V=1$ . Para medir el tono, se usa el ángulo alrededor del eje S. El rojo se sitúa a  $0^\circ$ , el verde a los  $120^\circ$  y el azul a los  $240^\circ$ . Los colores complementarios se encuentran a  $180^\circ$  de su color primario. El rango de S se extiende desde 0, situado en el eje de la pirámide, donde se sitúan los colores más oscuros, hasta 1, coincidiendo con el final del área hexagonal de la pirámide.

El vértice corresponde al negro con coordenadas  $S=0$  y  $V=0$ . El blanco corresponde a  $S=0$  y  $V=1$ . Los valores que se encuentran en el eje de la pirámide son los grises. Cuando  $S=0$  el valor de H no está definido. Sin embargo, a medida que S va creciendo, el valor de H comienza a tener importancia. Por ejemplo, el rojo puro se sitúa a  $H=0$ ,  $S=1$  y  $V=1$ . Si se añade blanco disminuye S, pero no cambia el valor de V. Las sombras se crean manteniendo  $S=1$  y disminuyendo V.

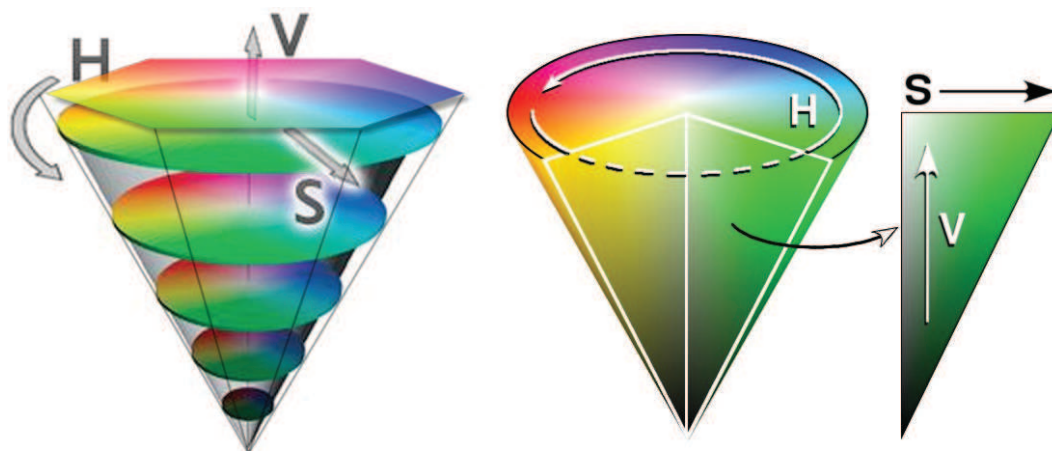


Figura 3.7. Representación gráfica del modelo HSV [44]

Este espacio se obtiene a partir de una transformación no lineal del espacio RGB, usando las siguientes relaciones:

$$H = \arccos \frac{\frac{1}{2}((R-G) + (R-B))}{\sqrt{((R-G)^2 + (R-B)(G-B))}}$$

$$S = 1 - 3 \frac{\min(R, G, B)}{R + G + B}$$

$$V = \frac{1}{3}(R + G + B)$$

El espacio HSV tiene la ventaja de ser invariante a las condiciones de luz; sin embargo, su alta complejidad computacional, sobre todo para sistemas basados en FPGA, lo convierte en un recurso de difícil implementación.

### 3.1.4.3 El modelo HSI.

El modelo de color HSI (del inglés Hue-Saturation-Intensity, Tono, Saturación, Intensidad) se define a través de una transformación no lineal del espacio de color RGB [44]. Esta transformación modifica el subespacio del cubo de la **Figura 3.6** convirtiéndolo en dos conos unidos por la base, tal como se muestra en la **Figura 3.8**. Geométricamente, la componente de saturación (componente S) se corresponde con la distancia radial de dicho cono, proporcionando una medida del grado en el que un color está mezclado con la luz blanca. Por otra parte, el tono (componente H) corresponde al ángulo respecto al eje rojo, proporcionando una magnitud de la longitud de onda dominante. El tono varía en un rango de  $360^\circ$  con una separación angular de  $120^\circ$  entre cada uno de los colores primarios; separación angular que también se mantiene entre los colores secundarios. Como se muestra en la figura, cuando  $H = 0^\circ$ , el color representado es el rojo, mientras que cuando  $H = 60^\circ$ , el color que se representa es el amarillo, y así sucesivamente.

La componente de intensidad "I" se obtiene como la distancia a lo largo del eje perpendicular al plano del color, la cual indica el valor del brillo del color y en consecuencia la información acromática relacionada con éste. Valores bajos de "I" corresponden a colores oscuros, mientras que valores superiores corresponden a colores claros hasta llegar al blanco.

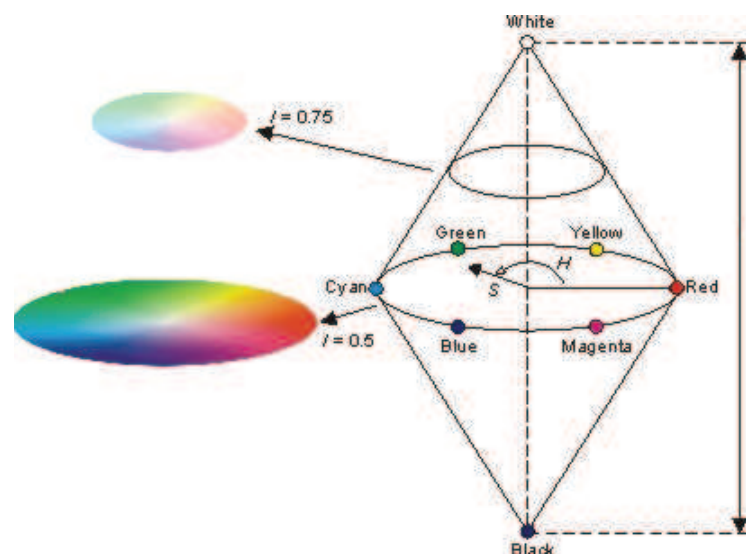


Figura 3.8. Representación gráfica del modelo HSI [44]

La similitud de las componentes de tono, saturación y brillo con la forma que tenemos los humanos de percibir el color, en la que existe una clara separación entre cromaticidad y acromaticidad, hacen que el modelo de color HSI sea una potente herramienta en el desarrollo de algoritmos de procesamiento digital de imagen [44].

$$H = \begin{cases} \theta & \text{si } B \leq G \\ 360 - \theta & \text{si } B > G \end{cases}$$

$$\theta = \cos^{-1} \left\{ \frac{(R - G) + (R - B)}{2[(R - G)^2 + (R - B)(G - B)]^{\frac{1}{2}}} \right\}$$

$$S = 1 - \frac{3 \min[R, G, B]}{(R + G + B)}$$

$$I = \frac{1}{3}(R + G + B)$$

A pesar de ello, al igual que el caso del Espacio HSV, la complejidad computacional de la transformación no lineal hace que su implementación sea poco práctica, a pesar de sus ventajas.

#### 3.1.4.4 El modelo YCbCr

YCbCr es una codificación no lineal del espacio de color RGB, usada comúnmente por los estudios de televisión europeos y en la compresión de imágenes. En el modelo YCbCr, el color se representa por tres parámetros, que son la luminancia (Y) y dos valores diferentes de color (Cb y Cr) que son características del color. La luminancia es la cantidad lineal de luz, que es directamente proporcional a la intensidad física. La luminancia aparece ponderada por la sensibilidad de la percepción humana con respecto al espectro visible y puede ser calculada como una suma ponderada de los componentes lineales del espacio de color RGB.

La obtención de este espacio de color a partir del RGB es la siguiente:

$$Y = 0.299R + 0.587G + 0.114B$$

$$Cr = R - Y$$

$$Cb = B - Y$$

Siendo R, G y B los valores del canal rojo, verde y azul respectivamente. La sencillez de la transformación y la separación explícita de las componentes de luminancia y de crominancia del color, hacen de este modelo un método atractivo en el procesamiento digital de imágenes. En particular, esta separación proporciona una clara ventaja en cuanto a la transferencia y compresión de los datos de color, ya que la mayor parte de la información se concentra en la componente de luminancia Y, que se transmite en alta precisión, reduciendo los valores Cb y Cr con métodos de compresión y sub-sampling incurriendo en muy poca pérdida de información.

En el caso de sistemas digitales, la ventaja del espacio YCbCr sólo abarca el ahorro de recursos, líneas y ancho de banda a transmitir, pero no proporciona una ventaja clara en los algoritmos de visión como por ejemplo la segmentación por color.

#### 3.1.4.5 Otros Modelos

Existen otros modelos ampliamente usados en el procesamiento de imágenes, como pueden ser el **HLS** (posteriormente mejorado y llamado **IHLS**), usado para la segmentación por

color basado en un umbral proporcional a la distancia euclidiana entre dos colores [45]. También es conocido el modelo  $L^*a^*b^*$ , que representa todos los colores del espectro visible (característica que el modelo RGB no posee), y representa la separación de los colores de modo proporcional a las diferencias visuales existentes entre ellos.

#### 3.1.4.6 Ventajas e inconvenientes de los diferentes Espacios de Color

La representación RGB de los colores se aleja mucho del concepto humano de color. Más aún, el procesamiento de imágenes en el modelo RGB tiene numerosas desventajas:

- Las tres componentes (R, G, B) dependen fuertemente de la intensidad.
- Los colores que el ojo humano percibe como colores cercanos, no tienen por qué estar cercanos en el Espacio RGB (en distancia euclídea).
- Las superficies de un solo color que están sombreadas suavemente podrían contener colores de muchos grupos distintos, haciendo difícil su caracterización.

Estas desventajas hacen que la umbralización por color, paso esencial en el proceso de detección de objetos se vea afectado en gran manera por los cambios en las condiciones de luz. Una posible solución sería el ajuste dinámico del valor del umbral dependiendo de las condiciones de luz. Otra solución sería cambiar a espacios de color como el HSI cuyos valores de color son robustos a los cambios de intensidad luminosa.

## 3.2 Análisis y procesamiento de imágenes

En este apartado se verán los conceptos básicos de análisis de imágenes y se introducirán los tipos de procesado más comunes.

El **Análisis Digital de Imágenes** es el área de la ingeniería que se encarga de la extracción de mediciones, datos o información contenida en una imagen. Incluye aquellas técnicas cuyo principal objetivo es facilitar la búsqueda e interpretación de la información contenida en ellas. Un sistema de análisis de imágenes se distingue debido a que tiene como parámetro de entrada una imagen, y cuyo resultado es comúnmente una salida numérica, en lugar de otra imagen. Esta salida es la información referente al contenido de la imagen de entrada [46].

Sin embargo, para llegar desde la imagen original al conjunto de parámetros e información extraída de la misma, es necesario pasar por distintas etapas de **procesamiento y filtrado** donde se analiza la imagen y se adecua para cierta aplicación específica. Esto implica que el resultado del procesamiento depende fuertemente del problema que se esté abordando [41].

El procesamiento y análisis de imágenes se ha desarrollado en respuesta a tres de los más grandes problemas concernientes a imágenes [47]:

- La digitalización y codificación de imágenes que facilite la transmisión, representación y almacenamiento de las mismas.
- Mejora y restauración de una imagen para interpretar más fácilmente su contenido.
- Descripción y segmentación de imágenes para aplicaciones de visión robótica o visión artificial.

Todos aquellos algoritmos de procesamiento de imágenes destinados a resaltar, agudizar y/o contrastar determinados aspectos de la imagen, y también aquellos que ayudan a eliminar



efectos no deseados sobre ellas, como toda clase de ruido (aditivo, sustractivo, multiplicativo, etc.), se denominan técnicas de mejora de la imagen [48].

El conjunto de métodos de procesamiento de imágenes está dividido en tres grandes grupos:

- **Algoritmos en el dominio espacial.** Se refiere a métodos que procesan una imagen píxel por píxel, o también tomando en cuenta un conjunto de píxeles vecinos.
- **Algoritmos en el dominio de la frecuencia.** Frecuentemente, estos métodos son aplicados sobre los coeficientes resultantes de la Transformada de Fourier de una imagen.
- **Algoritmos de extracción de características.** A diferencia de los dos grupos anteriores, los algoritmos de extracción de características están enfocados al análisis de imágenes para la extracción de atributos y regiones de interés, separación de objetos del fondo, detección de bordes o formas, entre otros.

### 3.2.1 Métodos en el dominio espacial

Incluyen todos los métodos que se basan en el procesado de un píxel (llamado píxel actual) a partir de una relación que puede incluir a los píxeles vecinos.

#### 3.2.1.1 Relaciones entre píxeles vecinos

Un píxel  $p$  situado en un plano del espacio cuyas coordenadas son  $(x, y)$  tiene 4 vecinos horizontales y 4 verticales, cuyas coordenadas están dadas por:

$$(x + 1, y), (x - 1, y), (x, y + 1), (x, y - 1)$$

Este grupo de píxeles se nota como  $N4(p)$ . Así mismo, las vecindades diagonales con el punto  $(x, y)$  se notan como  $ND(p)$ , y sus coordenadas son:

$$(x + 1, y + 1), (x - 1, y + 1), (x - 1, y - 1), (x + 1, y - 1)$$

El conjunto del píxel  $p$  y sus vecinos crea una región. Esta región establece que dos píxeles son adyacentes si, y solo si, tienen en común una de sus fronteras, o al menos una de sus esquinas. La **Figura 3.9** muestra la relación de vecindad de un píxel  $p$ . El conjunto de píxeles vecinos al píxel actual suele llamarse *ventana* o *plantilla* [50].

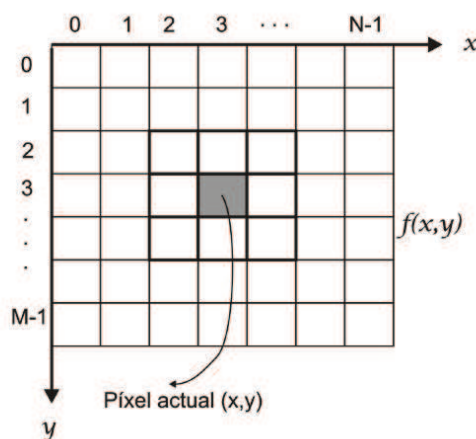


Figura 3.9. Vecinos del píxel actual 3x3.

### 3.2.1.2 Tipos de transformaciones

Según la relación del píxel de salida con los vecinos del píxel actual, las transformaciones de una imagen de entrada en una imagen procesada pueden clasificarse de la siguiente forma:

- **Transformaciones puntuales.** Son aquellas en las cuales el píxel resultante de la operación depende sólo del valor del píxel de entrada. Las operaciones puntuales típicas incluyen la manipulación de los píxeles uno a uno, por ejemplo la binarización, la segmentación, la corrección de color, tono, saturación, gamma, etc.
- **Transformaciones locales.** En este caso, para obtener el píxel de salida, se utilizan las contribuciones de los píxeles vecinos en la operación. Muchas operaciones son locales, por ejemplo, suavizado, media, operaciones morfológicas, realce de bordes. Se clasifican en filtros lineales, como la media, y los no lineales, como la mediana.
- **Transformaciones globales.** El píxel de salida como resultado de la operación, se obtiene a partir del total de datos de la imagen como valor de entrada. Las operaciones globales se realizan a menudo en el dominio de la frecuencia. Un ejemplo es la compresión de imágenes que tomando el total de una imagen entrada obtiene una imagen comprimida de salida.
- **Transformaciones geométricas.** Se realizan tomando en cuenta las posiciones de los píxeles en la imagen, y se les aplican operaciones de translación / rotación. Ejemplos típicos son rotación, traslación, cambios de escala, rectificación, y transformaciones radiométricas de los píxeles.

Además, si la imagen a tratar es binaria (donde todos los píxeles toman un valor '0' o '1' según pertenezcan al fondo o a objetos en primer plano), se definen las transformaciones lógicas, donde el píxel de salida es el resultado de aplicar operadores lógicos (AND, OR, XOR) sobre dos imágenes binarias (**Figura 3.10**).

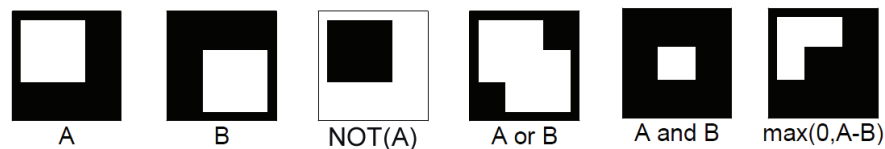


Figura 3.10. Operaciones lógicas sobre imágenes binarias.

Por último, las ventanas usadas en las operaciones locales no están limitadas solamente a los píxeles adyacentes ( $N_4(p)$ ,  $N_D(p)$ ), es decir, no tienen por qué tener un tamaño de  $3 \times 3$ . Un claro ejemplo lo encontramos en el filtro media, que suaviza la imagen, reduciendo la cantidad de variaciones de intensidad entre píxeles vecinos (**Figura 3.11**); cuanto más grande sea la ventana del filtro, mejor serán los resultados de la imagen obtenida.

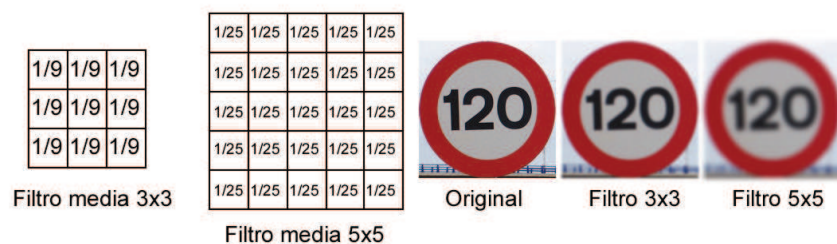


Figura 3.11. Diferentes tamaños de la máscara del filtro media y sus resultados sobre una imagen.

Las operaciones espaciales toman como entrada una imagen, y recorren cada uno de sus píxeles, utilizando para ello una ventana de vecindades de tamaño  $N \times N$ . De esta forma procesan una imagen recorriendo todos sus elementos, y aplicando una transformación sobre ellos, esto es:

$$g(x, y) = T[f(x, y)]$$

donde  $f(x, y)$  es la imagen de entrada,  $g(x, y)$  es la imagen procesada o resultante, y  $T$  es un operador que se aplica sobre la imagen, el cual es definido sobre los vecinos del píxel  $(x, y)$ . Si sólo se tiene en cuenta el píxel actual, el operador  $T$  sería una matriz de tamaño  $1 \times 1$ , tomando de esta forma un valor constante. En este caso,  $g(x, y)$  dependería únicamente de  $(x, y)$ , que es el píxel que se está procesando. Con ello se obtienen todos los tipos de transformaciones espaciales vistos anteriormente. Así, si  $T$  es un valor escalar, la transformación será puntual, y en otro caso será local o global.

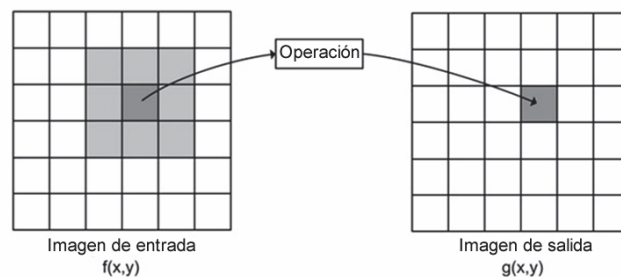


Figura 3.12. Transformación espacial sobre una imagen.

### 3.2.1.3 Transformaciones de suavizado

Esta técnica sirve para suavizar los bordes de una imagen, reducir los picos de ruido o simplemente hacer menos bruscos los cambios de intensidad en la imagen. Se consideran filtros paso-bajo, ya que eliminan la información de alta frecuencia en la imagen.

**Filtro Media.** El filtro de suavizado más simple es el correspondiente a la media, en la cual se obtiene un píxel de salida haciendo la media aritmética de los píxeles de la ventana elegida (**Figura 3.11**). Cuanto mayor sea la ventana, mayor será el efecto de difuminado. A pesar de su sencillez en la implementación, tiene como desventaja su alta sensibilidad a los cambios locales, y la generación de niveles de intensidad que no existían en la imagen original.

**Filtro Gaussiano.** Similar al filtro media, pero aplicando una máscara diferente, donde los píxeles más cercanos al píxel actual tienen más peso que los exteriores. Los pesos de los píxeles se calculan con una campana de Gauss dependiente de la distancia al píxel actual, y su varianza indica el nivel de suavizado. Los filtros gaussianos tienen como ventaja el hecho de ser separables, es decir, se pueden realizar con la convolución de dos vectores unidimensionales en lugar de una máscara bidimensional. También mejoran la capacidad de suavizado, introduciendo un parámetro (la varianza) que es independiente del tamaño de la máscara.

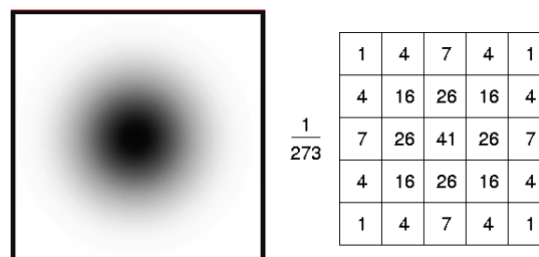


Figura 3.13. Máscara de un filtro gaussiano 5x5 con varianza 1.

### 3.2.1.4 Transformaciones de perfilado

Las transformaciones de perfilado realizan la operación contraria al suavizado. Sirven para destacar y hacer más visibles las variaciones y bordes de la imagen. El perfilado mejora la apariencia difuminada de las imágenes, que puede ser debida a imperfecciones en las lentes u otra causa.

El perfilado suele realizarse sumando a la imagen original el operador Laplaciano ponderado por cierto factor. El Laplaciano de una imagen se define como:

$$L(x, y) = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}$$

Donde el valor  $I$  es la intensidad del píxel actual, suponiendo una imagen en escala de grises, y los valores  $x, y$  corresponden a la posición del píxel actual con respecto a sus vecinos. Este operador hace uso de la segunda derivada espacial de la imagen para destacar las regiones donde existen cambios bruscos de intensidad, y por tanto también se lo utiliza para la detección de bordes.



Figura 3.14. Máscara 3x3 del Laplaciano y resultados de perfilado.

### 3.2.1.5 Detección de bordes

La detección de contornos es una práctica básica en el procesamiento de imágenes, pues proporciona información útil acerca de los límites del objeto que pueden ser utilizados para el análisis, detección del objeto y para aplicaciones de filtrado. De igual forma se emplea para simplificar el análisis de imágenes, realizando una reducción drástica de la cantidad de datos a ser procesados, mientras que al mismo tiempo preservan la información estructural alrededor de los límites del objeto [50].

La detección de bordes tiene como objetivo detectar las zonas de cambios bruscos de intensidad y realzarlas, obteniendo una imagen resultante en la cual se han destacado los bordes. Son muy efectivos para acentuar el contraste y detectar puntos aislados o pequeños detalles, si se aumenta el tamaño de la máscara. De la misma forma es posible diseñar máscaras de detección de bordes que detecten la presencia de líneas finas en una imagen.

La característica común a todos los filtros de detección de bordes es la combinación de pesos positivos con negativos, que no son más que la forma discreta de las derivadas (gradientes) espaciales de la imagen. Al usar el concepto de gradiente, la dirección espacial en la cual la variación de intensidad es mayor, se establece una imagen de salida donde se realzan las curvas en las cuales el gradiente es máximo.

Los bordes de una escena suelen ser invariantes a los cambios de luz. Es por ello que muchos sistemas de análisis de imágenes utilizan detección de bordes para trabajar con éstos, y no con la imagen original. Sin embargo, los filtros de detección de bordes son muy sensibles al ruido, por lo cual es habitual combinarlos con filtros de suavizado previos a la etapa de detección. Los más comunes se mencionan a continuación [49].

**Filtros de Prewitt.** Basados en la estimación del módulo del gradiente usando máscaras 3x3. El operador de Prewitt otorga el mismo peso a los píxeles contiguos en vertical y horizontal, que a los contiguos en diagonal.

**Filtros de Sobel.** Se construyen usando la derivada de la Gaussiana. Permite además calcular derivadas conjuntas en X e Y, derivadas segundas, terceras, etc.

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

Figura 3.15. Máscaras del operador de Sobel con derivadas en X (izquierda) y en Y (derecha).

El operador Sobel calcula el gradiente de la intensidad de una imagen en cada píxel. Así, para cada punto, este operador proporciona la magnitud del gradiente, su dirección y sentido desde el más oscuro al más claro. La ventaja adicional que presentan estas máscaras sobre las anteriores es que además de estimar el valor del módulo del gradiente, al derivar sobre la Gaussiana, producen un alisamiento en la imagen que es beneficioso, dado el comportamiento ruidoso que presentan las estimaciones basadas en derivadas.

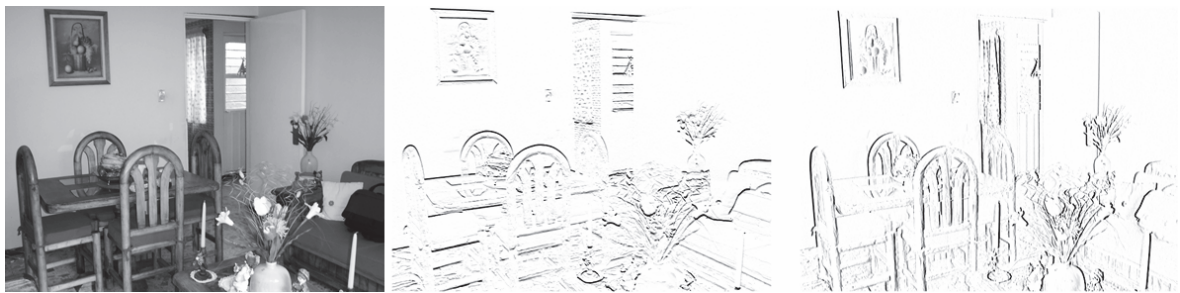


Figura 3.16. A la izquierda la imagen original. En el centro y en la derecha, la aplicación de la máscara con derivadas en X e Y respectivamente [169].

**Detector de bordes de Canny.** No sólo usa operadores de gradiente, sino que busca el máximo gradiente a lo largo de un borde. El resultado es una imagen binaria (borde/no borde), ajustable mediante un umbral.

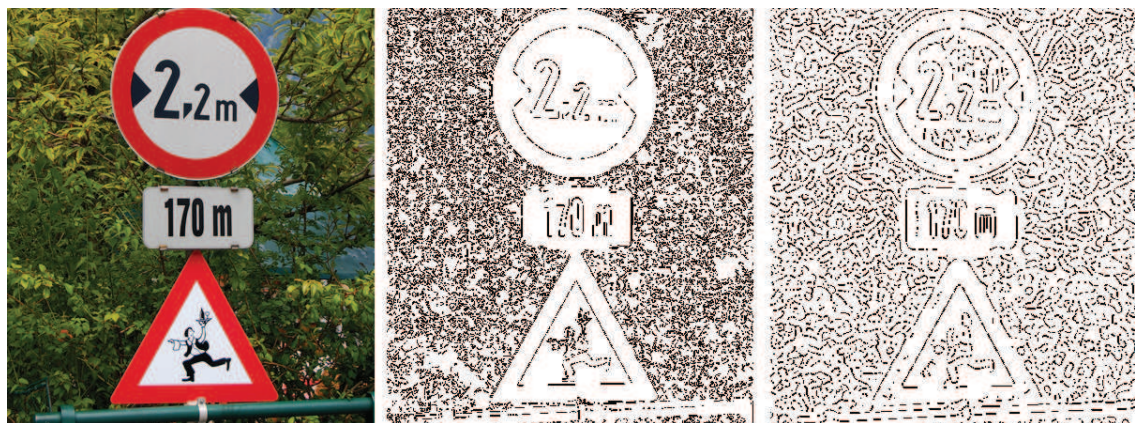


Figura 3.17. Imagen original y aplicación de un detector de bordes de Canny con umbrales 0.5 (centro) y 0.1 (derecha).

A continuación se muestra un resumen de los principales métodos de detección de bordes basados en el gradiente, junto a sus ventajas principales [51].

<b>Métodos basados en la primera derivada:</b> Operador Gradiente. Constituyen los métodos con más proliferación dentro de la Comunidad del Análisis de Imagen y la Visión Computacional. Se fundamentan en que un borde existe si hay una discontinuidad en la función de intensidad de la imagen, es decir, si la derivada de los valores de intensidad de la imagen es un máximo.		
<b>Operadores gradiente</b>	<b>Ventajas</b>	<b>Desventajas</b>
<p><b>Prewitt</b></p> $\frac{1}{3} \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} \quad \frac{1}{3} \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$	<ul style="list-style-type: none"> <li>• Buena respuesta en bordes horizontales y verticales.</li> <li>• Poco sensible al ruido.</li> <li>• Proporciona la magnitud y dirección del borde.</li> </ul>	<ul style="list-style-type: none"> <li>• Mala respuesta en bordes diagonales.</li> <li>• Lentitud de cálculo.</li> <li>• Anchura del borde de varios píxeles.</li> </ul>
<p><b>Sobel</b></p> $\frac{1}{4} \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \quad \frac{1}{4} \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$	<ul style="list-style-type: none"> <li>• Buena respuesta en bordes horizontales y verticales.</li> <li>• Diversidad de tamaños en las máscaras.</li> <li>• Poco sensible al ruido.</li> </ul>	<ul style="list-style-type: none"> <li>• Mala respuesta en bordes diagonales.</li> <li>• Lentitud de cálculo.</li> <li>• No da información acerca de la orientación del borde.</li> <li>• Anchura del borde de varios píxeles.</li> </ul>
<p><b>Roberts</b></p> $\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix} \quad \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	<ul style="list-style-type: none"> <li>• Buena respuesta en bordes horizontales y verticales.</li> <li>• Buena localización.</li> <li>• Simpleza y rapidez de cálculo.</li> </ul>	<ul style="list-style-type: none"> <li>• Mala respuesta en bordes diagonales.</li> <li>• Sensible al ruido.</li> <li>• Empleo de máscaras pequeñas.</li> <li>• No da información acerca de la orientación del borde.</li> <li>• Anchura del borde de varios píxeles.</li> </ul>
<p><b>Desventajas generales de la detección de bordes basada en el Gradiente</b></p> <ul style="list-style-type: none"> <li>• La fijación por parte del usuario de los umbrales y el tamaño de la máscara, afectará a la posición del borde.</li> <li>• El gradiente presenta una excesiva dependencia con respecto a la dirección de barrido, por ello, las aristas cuyas pendientes están próximas a la dirección de barrido no se detectan fácilmente.</li> <li>• La debilidad del Gradiente en los puntos esquina provocará la pérdida de puntos relevantes y marcado de juntas.</li> </ul>		

Tabla 3.1. Métodos más comunes de detección de bordes basados en el gradiente [51].

Los distintos operadores presentados hasta ahora son operadores de primeras derivadas, es decir, basados en el operador gradiente. Esto implica que dependen fuertemente de la dirección del borde a detectar. Si se desea detectar bordes en todas las direcciones posibles, se deben utilizar diferentes máscaras. Por ejemplo, el operador gradiente tiene dos máscaras y detecta discontinuidades en filas ó columnas.

Sin embargo, pueden existir aplicaciones en las cuales se necesite utilizar un operador que sea independiente de la orientación de los bordes. Uno de ellos es el operador de **Laplace ó Laplaciano**. Puesto que el Laplaciano es un operador de segundas derivadas, es mucho más sensible al ruido que los operadores anteriormente descritos. Al igual que con el operador Gradiente existen multitud de ventanas correspondientes a operadores Laplacianos, se describen a continuación las más populares [51].

<p><b>Métodos basados en la segunda derivada:</b> Operador Laplaciano. Constituyen operadores ideales para detectar bordes independientemente de la orientación o dirección de los mismos. Se fundamentan en que cuando la imagen presenta un cambio de intensidades a lo largo de una determinada dirección, existirá un máximo en la primera derivada a lo largo de dicha dirección y un paso por cero en la segunda derivada.</p>		
<p><b>Operadores Laplacianos</b></p> $\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$ $\frac{1}{4} \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad \frac{1}{4} \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$ $\frac{1}{3} \begin{bmatrix} 2 & -1 & 2 \\ -1 & -4 & -1 \\ 2 & -1 & 2 \end{bmatrix} \quad \frac{1}{3} \begin{bmatrix} -2 & 1 & -2 \\ 1 & 4 & 1 \\ -2 & 1 & -2 \end{bmatrix}$	<ul style="list-style-type: none"> <li>• Buena localización siempre que las aristas estén bien separadas y la relación señal ruido sea alta.</li> <li>• Anchuras de bordes óptimas.</li> <li>• Independiente de la orientación del borde: buena respuesta en bordes horizontales, verticales y diagonales.</li> </ul>	<ul style="list-style-type: none"> <li>• Muy sensible al ruido.</li> <li>• Fiabilidad baja, pudiendo aparecer como identificados falsos bordes.</li> </ul>

Tabla 3.2. Métodos más comunes de detección de bordes basados en la segunda derivada o Laplaciano [51].

Finalmente, existen otros operadores de detección de bordes más complejos, como son los operadores DoG (Derivada de la gaussiana) o el operador LoG (Laplaciano del Gaussiano), que no son más que la convolución previa de una máscara de suavizado con los operadores Gradiente o Laplaciano.

### 3.2.1.6 Transformaciones no lineales

Los más comunes son los filtros estadísticos de orden, que son aquellos que toman los valores en la vecindad de cada punto y los ordenan de menor a mayor, obteniendo algún valor de salida a partir de la lista ordenada.

Este tipo de filtros son muy usados para encontrar los puntos más nítidos o más oscuros de una imagen y resaltarlos, o para eliminar ciertos tipos de ruido de alta frecuencia como el ruido sal y pimienta, en el cual ciertos píxeles de una imagen son muy diferentes en color o en intensidad a los píxeles circundantes. Los tipos de filtros no lineales más conocidos son la Mediana, el Mínimo y el Máximo.

**Filtro de mínimo.** Selecciona el menor valor dentro de una ventana ordenada de valores de nivel de gris (o intensidad de color en cada componente, en caso de una imagen a color). Elimina el ruido tipo sal (píxeles blancos), aunque como desventaja tiende a oscurecer la imagen.

**Filtro de máximo.** Selecciona el valor más grande dentro de una ventana ordenada de valores de nivel de gris (o intensidad de color en cada componente, en caso de una imagen a

color). Elimina el ruido tipo pimienta (píxeles negros), aunque como inconveniente, tiende a aclarar la imagen.

**Filtro mediana.** Realiza una operación estadística de mediana con los píxeles de la ventana, ordenándolos en primer lugar de menor a mayor intensidad, para luego tomar el valor que esté en medio y sacarlo como píxel de salida. Este filtro no es lineal, por lo cual dadas dos imágenes A y B, la mediana de A+B no da el mismo resultado que la mediana de A más la mediana de B. Como ventaja, este filtro sirve para eliminar el ruido "sal y pimienta", como se ve en la **Figura 3.18**.

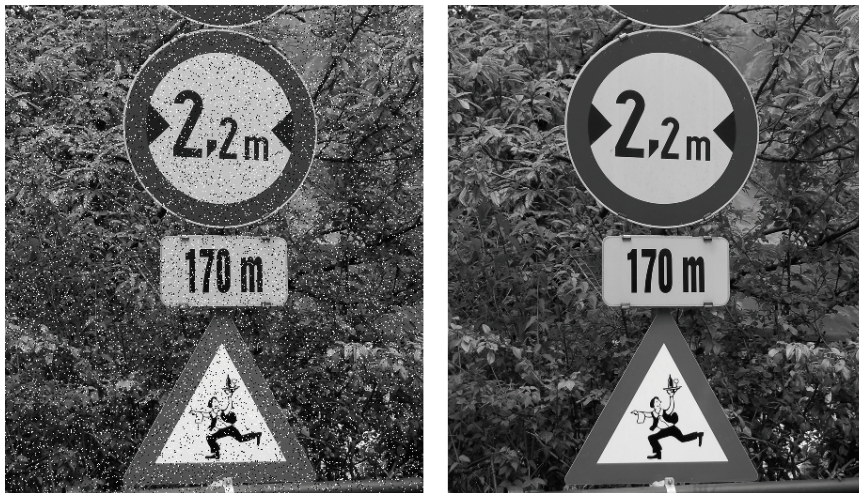


Figura 3.18. Imagen con ruido tipo "sal y pimienta" y aplicación del filtro mediana.

Al igual que los filtros lineales, el resultado de la aplicación del filtro mediana depende de la relación señal-ruido, que dará mejores resultados cuanto mayor sea este parámetro.

### 3.2.1.7 Operaciones morfológicas

Las operaciones morfológicas son métodos que tienen su origen en la teoría de conjuntos. En el procesamiento de imágenes, se suele aplicar sobre imágenes binarias, donde se ha hecho una segmentación previa, separando el fondo (marcado como '0') de los objetos de interés (marcados como '1'). Una imagen binaria es un conjunto de valores organizados en una cuadrícula, en la cual cada píxel sólo puede tener dos valores, 0 ó 1. Como es lógico suponer, al tener una imagen en esas condiciones es mucho más fácil encontrar y distinguir características estructurales.

Las operaciones morfológicas procesan estas imágenes binarias basándose en la forma de sus objetos de interés. En general, toman una imagen binaria como entrada y dan como resultado otra imagen binaria. El valor de cada píxel en la imagen de salida se obtiene con operaciones no lineales sobre el píxel de entrada y sus vecinos. En general, las operaciones morfológicas se usan para:

- Supresión de ruidos.
- Simplificación de formas.
- Destacar la estructura de los objetos (detección de envolvente, ampliación, reducción).
- Descripción de objetos (área, perímetro).

Las dos operaciones morfológicas más conocidas son la erosión y la dilatación, cuyo objetivo principal es simplificar las imágenes para un posterior análisis, conservando todas sus características.



**Dilatación.** Dada una imagen binaria A (que en rigor es un conjunto de la cuadrícula  $\mathbb{Z}^2$ , y un elemento estructural B, la dilatación de A por B se define como:

$$A \oplus B = \{x \mid (\hat{B})_x \cap A \neq \emptyset\}$$

Que se entiende como "aquellos píxeles x tales que la intersección de la estructura B situada sobre x y la imagen A es distinto del conjunto vacío". Teniendo en cuenta que para la intersección de A y B sólo se toman en cuenta los píxeles que correspondan a los objetos de primer plano (píxeles a '1'). El elemento estructural B indica de qué forma se llevará a cabo la dilatación.

La operación de dilatar una imagen se puede describir como un crecimiento (o "dilatación") de los píxeles situados alrededor de los bordes de los objetos. En general, este método marca como '1' todos los píxeles que formen parte del fondo de la imagen, pero que al mismo tiempo estén en contacto directo con el objeto. Esto permite aumentar en uno el nivel de píxeles en el perímetro de cada objeto, que sufre un crecimiento de tamaño, y al mismo tiempo permite rellenar posibles huecos dentro del mismo.

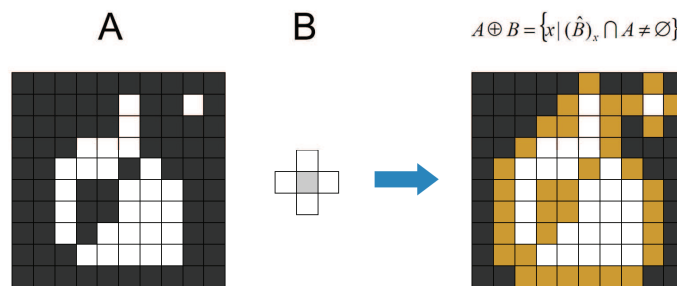


Figura 3.19. Resultados de la dilatación sobre imágenes binarias.

El estado de cualquier píxel de salida es obtenido aplicando una regla determinada al píxel de entrada y a sus vecinos. La operación realizada para obtener una imagen dilatada es la siguiente: "Si cualquier píxel vecino del píxel de entrada es '1', entonces el píxel de salida es también '1'. En cualquier otro caso el píxel de salida será '0'".

**Erosión.** Dada una imagen binaria A (que en rigor es un conjunto de la cuadrícula  $\mathbb{Z}^2$ , y un elemento estructural B, la erosión de A por B se define como:

$$A \ominus B = \{x \mid B_x \subseteq A\}$$

Que se entiende como "aquellos píxeles x tales que la estructura B situada sobre x pertenezca en su totalidad a la imagen A". Teniendo en cuenta que para la condición "Bx pertenece a A" se toman sólo los píxeles de los objetos en primer plano (aquellos marcados como '1'). La erosión es la operación morfológica dual a la dilatación y se concibe usualmente como una reducción de la imagen original.

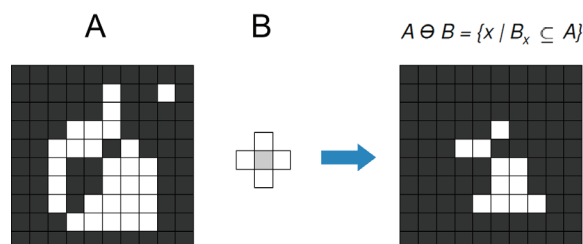


Figura 3.20. Resultados de la erosión sobre una imagen binaria.

El estado de cualquier píxel de salida es obtenido aplicando una regla determinada al píxel de entrada y a sus vecinos. La operación realizada para obtener una imagen erosionada es la siguiente: "Si todos los píxeles vecinos del píxel de entrada están a '1', entonces el píxel de salida será '1'. En cualquier otro caso, el píxel de salida será '0'."

Aplicando en conjunto estas dos operaciones, erosión y dilatación, se obtienen interesantes resultados en el análisis de imágenes, suavizando los contornos, rellenando huecos para hacer los objetos más homogéneos, eliminando ruido y puntos de tamaño demasiado pequeños para resultar de interés, etc. (**Figura 3.21**). Nótese que al aplicar estas dos operaciones morfológicas en cadena, los objetos de mayor tamaño han quedado invariantes, mientras que han desaparecido los objetos más pequeños. Variando el elemento estructural B, se puede indicar el tamaño mínimo de los objetos para que se vean inalterados, eliminando el resto.

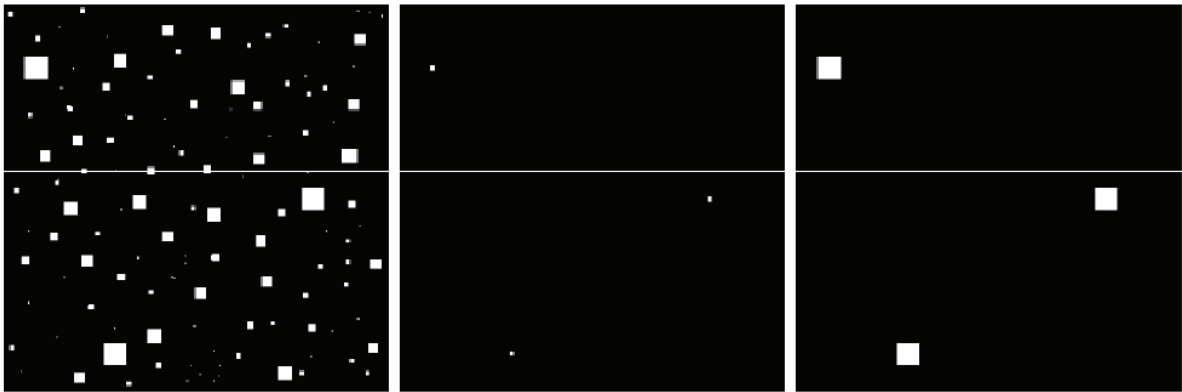


Figura 3.21. Resultado de aplicar erosión seguida de dilatación en una imagen binaria.

Por último, existen otros tipos de operaciones morfológicas, aunque no son de interés en este Proyecto Fin de Carrera, como por ejemplo la transformada **Hit-or-Miss** para adelgazar imágenes y detectar contornos, la **apertura** y la **clausura** para eliminar pequeños salientes y huecos, entre otros.

### 3.2.1.8 Transformaciones geométricas

Las operaciones geométricas modifican las relaciones espaciales existentes entre los píxeles de una imagen y pueden ser de tres tipos: traslación, escalado y rotación.

A diferencia de las operaciones descritas hasta ahora, las transformaciones geométricas cambian la proyección de la imagen sobre el plano que la contiene. La imagen resultante difiere en tamaño y quizás en forma con respecto a la original. Existen numerosos modelos matemáticos que permiten modificar las relaciones geométricas de una imagen, aunque en general una transformación geométrica puede expresarse como:

$$x' = S \cdot R \cdot x + t$$

Donde  $x$  es el vector que corresponde a un píxel de la imagen de entrada,  $x'$  es el vector resultante tras aplicar la transformación,  $S$  es un factor de escala aplicado sobre la transformación,  $R$  una matriz de rotación, que puede o no ser ortogonal, y  $t$  es el vector utilizado para operaciones de traslación.

Este tipo de transformaciones son muy importantes en el análisis de imágenes, sobre todo para el reconocimiento de patrones. Dependiendo de la posición de los objetos en el espacio al ser captados por una cámara pueden (y suelen) modificar su forma, transformando por ejemplo

los círculos en elipses, los cuadrados en formas romboides. Por ello en este tipo de situaciones, antes de comparar patrones y detectar formas, se precisa una etapa previa de transformación de la imagen.



Figura 3.22. Aplicación de transformaciones geométricas.

### 3.2.2 Métodos en el dominio de la frecuencia

Estos algoritmos se basan en filtros que procesan una imagen trabajando sobre el dominio de la frecuencia en la Transformada de Fourier de la imagen. Debido a que la imagen es considerada como una función de dos dimensiones finita y discreta, existe su Transformada Discreta de Fourier (DFT). Para obtener la DFT, se modifica la imagen original siguiendo el Teorema de la Convolución.

#### 3.2.2.1 Transformaciones en el dominio de Fourier

Una función de transferencia de filtrado  $H(u, v)$  es aquella que actúa sobre la transformada de Fourier de una imagen  $F(u, v)$ , y permite suprimir ciertas frecuencias mientras deja otras sin cambio alguno. Las frecuencias bajas son responsables de la mayoría de los niveles de gris de una imagen sobre áreas suaves. Mientras que las frecuencias altas tienen que ver con los detalles de la imagen, como son los bordes y el ruido [41], [52]. La realización de filtros en el dominio de la frecuencia para mejoras en las imágenes requiere del proceso que se resume a continuación:

1. Multiplicación de la imagen digital  $f(x, y)$  de tamaño  $M \times N$  (en niveles de grises o por separado en cada plano de color) por un factor de  $(-1)^{x+y}$  con el fin de situar el cálculo de la transformada en la coordenada  $(M/2, N/2)$ . De esta forma se centra la transformada, siendo el valor de frecuencia cero  $F(0,0)$  el valor medio de gris de la imagen.
2. Calcular  $F(u, v)$  con la ayuda de un procesador, la DFT del paso anterior  $F(u,v) = \text{DFT}[f(x,y) \cdot (-1)^{x+y}]$ .
3. Multiplicar  $F(u, v)$  por la Transformada de Fourier de la función de transferencia del filtro,  $H(u, v)$ .
4. Calcular la DFT inversa del resultado del paso anterior.
5. Obtener la parte Real del paso anterior (la mayoría de filtros actúan en magnitud e ignoran la fase).
6. Multiplicar nuevamente el resultado anterior por el factor  $(-1)^{x+y}$ , para obtener la imagen final.

La **Figura 3.23** muestra el diagrama de bloques del procedimiento de filtrado en el dominio de la frecuencia, donde se incluyen las etapas de pre- y post-procesamiento. Las operaciones que realiza un procesador para llevar a cabo este proceso, en ocasiones pueden costarle demasiado tiempo y/o un excesivo uso de sus recursos. Cuando esto sucede, se prefiere llevar a cabo la especificación del filtro en el dominio de la frecuencia, para después implementarlo en el dominio espacial mediante la reducción del filtro a una máscara del orden de una matriz de  $3 \times 3$ , que se aplica sobre cada píxel de la imagen en forma de convolución.

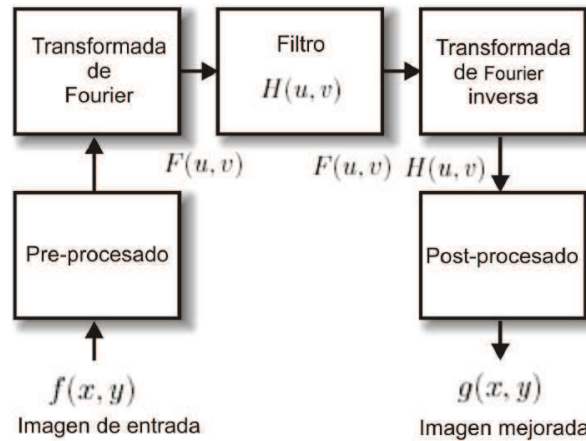


Figura 3.23. Proceso de filtrado en el dominio de la frecuencia.

Los filtros en el dominio de la frecuencia más importantes son el Butterworth y el Gaussiano, ambos en sus versiones paso-bajo y paso-alto.

### 3.2.2.2 Transformaciones basadas en Histograma

El histograma de una imagen es una representación gráfica de la frecuencia con la que los niveles de gris aparecen en ella (o niveles de intensidad en cada plano de color, en caso de una imagen en color). Es una herramienta fundamental para el análisis de imágenes digitales, ya que permite “condensar” mucha información sobre la imagen (probabilidades de cada nivel de gris) aunque se pierde la localización espacial. Su rango dinámico es el conjunto de niveles de gris presentes.

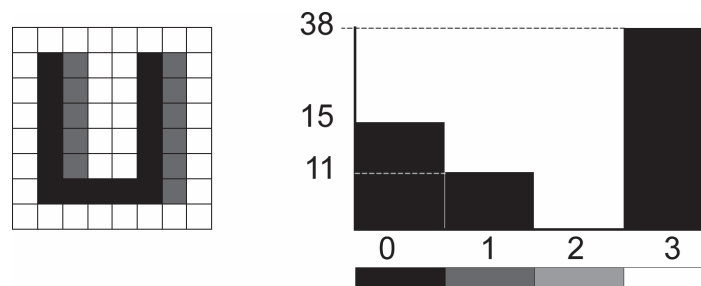


Figura 3.24. Ejemplo de imagen y su histograma.

Se suele representar como una gráfica de puntos, donde el eje horizontal representa todos los posibles valores de intensidad de un píxel, y el eje vertical la frecuencia de aparición de dichos valores en la imagen (**Figura 3.24**).

Las transformaciones de la imagen basadas en su histograma se pueden visualizar mediante funciones de transferencia, que corresponden a curvas en los ejes de abscisas y ordenadas, acotadas entre 0 y 1. Estas funciones de transferencia comprimen / expanden de forma independiente los ejes de coordenadas del histograma, y pueden ser lineales o de cualquier orden.

El histograma de una imagen es muy utilizado en el análisis de imágenes y en visión por computador, ya que permite operaciones de aumento de contraste, ecualización, o umbralización por niveles de intensidad. Un ejemplo de este último caso lo vemos en la **Figura 3.25**, donde se ha utilizado un histograma bimodal, en el que existen dos grandes agrupaciones de niveles que aparecen con cierta frecuencia en la imagen, una correspondiente al objeto oscuro y otra debida al fondo claro. Una vez determinadas estas dos agrupaciones, el nivel de gris equidistante entre ambas, resultaría un nivel de umbral adecuado para la separación del objeto del fondo.

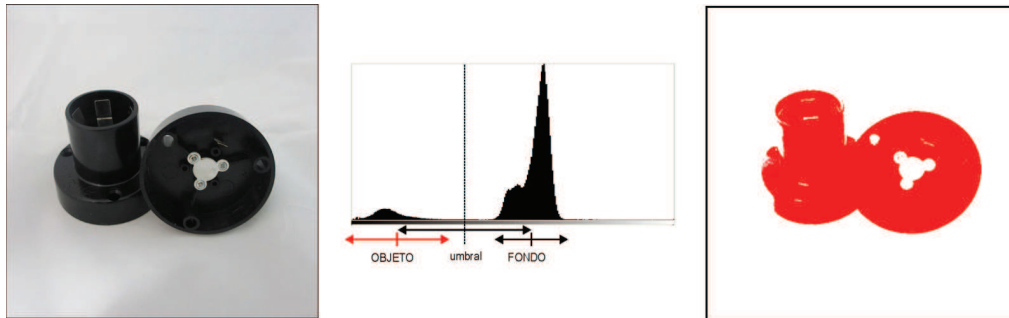


Figura 3.25. Umbralización de una imagen con ayuda del histograma.

### 3.2.3 Métodos de extracción de características

Al contrario que los métodos vistos hasta ahora, la extracción de características es un método que toma una imagen como entrada y extrae atributos de interés de la misma como pueden ser coordenadas de objetos que cumplan ciertas características, detección de curvas y formas, etiquetado de componentes, entre otros. La extracción de características entra de lleno en el campo del análisis de imágenes constituyendo la primera etapa en la inteligencia de un sistema de visión artificial.

La extracción de información de interés que contiene una imagen constituye actualmente un inmenso campo de estudio e investigación, que abarca multitud de aplicaciones. Las áreas que abarcan el estudio del análisis de imágenes son tan variadas como la informática, la física, la fotogrametría y las matemáticas, entre otros. Estas investigaciones abordan problemas que van desde la aplicación de simples filtros lineales hasta la automatización del reconocimiento semántico de objetos. La detección automática de características sobre imágenes en el campo de visión por computador tiene una dilatada tradición y cuenta multitud de métodos para tal propósito. Desafortunadamente, no existe un 'método universal' para la detección automática de características, sino que serán los requerimientos del propio problema los que obliguen a personalizar la metodología.

Una imagen contiene una gran cantidad de datos, pero por lo general, la mayoría de éstos proporcionan muy poca información útil para interpretar la escena. Un sistema de visión artificial necesita realizar un primer paso que consistirá en extraer de la forma más robusta, eficaz y rápida posible, las características de la escena que proporcionen la información que se necesita para un posterior paso de interpretación. Estos sistemas deben cumplir, entre otras, las siguientes condiciones:

- La extracción de información útil a partir de la imagen no debería suponer un coste excesivo al sistema en el cual está integrado, y el tiempo total dedicado a esta tarea debería ser lo menor posible.
- La localización de las características de la imagen debe ser muy precisa. Así mismo, el error cometido en la estimación de las mismas debe ser lo más pequeño posible.
- El método utilizado para la extracción de características debe ser robusto y estable.

- Los datos extraídos deberían contener la máxima información posible de la escena, llegando incluso a extraer la información geométrica contenida en la misma.

En este apartado se verán los tres métodos más comunes en la extracción de características: la segmentación, el etiquetado y la detección de formas.

### 3.2.3.1 Segmentación

La segmentación subdivide una imagen en sus regiones u objetos constituyentes, de tal manera que los píxeles de esas regiones posean propiedades o atributos similares, como niveles de gris, contraste o texturas.

La mayoría de los algoritmos de segmentación están basados en dos propiedades básicas de intensidad de la imagen: la discontinuidad y la similitud. En la categoría de segmentación mediante discontinuidad, el proceso se realiza dividiendo la imagen por cambios abruptos en intensidad, como es el caso de la detección de bordes en una imagen. Con respecto a la segmentación con base en la similitud, ésta se logra mediante la partición de una imagen en regiones que son similares de acuerdo a un conjunto de criterios predefinidos [41].

El proceso de segmentación se encarga de evaluar cada píxel de la imagen y decidir si contiene o no las características de interés. Como resultado, este método genera una imagen binaria, donde los píxeles que pertenecen al objeto se representan con un '1' (objeto en primer plano), mientras que los que no pertenecen al mismo se representan con un '0' (fondo). La decisión de pertenencia de un píxel a uno u otro segmento se basa en el análisis de alguna característica de la imagen, como por ejemplo los niveles de intensidad o la textura.

Existen diferentes tipos de segmentación, listados a continuación:

- **Segmentación basada en características de píxel**
  - Segmentación por niveles de gris
  - Segmentación de imágenes en color
  - Segmentación por texturas
- **Segmentación basada en transiciones**
  - Detección de bordes
- **Segmentación basada en modelos**
  - Transformada de Hough
- **Segmentación basada en homogeneidad**
  - Fusión de regiones
  - Zonas planas
  - Propagación de Marcadores
- **Segmentación basada en Morfológica Matemática**

Las técnicas de segmentación dependen fuertemente del objetivo que persigue la aplicación en particular, así como del tipo de imagen a analizar y sus características. Por lo tanto, en una etapa previa a la segmentación, es preciso tener claro qué objetos interesan y qué características poseen. También es común realizar operaciones de filtrado una vez terminada la etapa de segmentación, así como determinar las características que permitan separar y clasificar los objetos encontrados.

Debido al gran número de imágenes y aplicaciones diferentes que existen, es difícil evaluar la eficacia de un método de segmentación para una aplicación específica. Fundamentalmente, lo que se busca es que diferentes objetos tengan valores claramente diferentes de las

características discriminantes, y el éxito de la operación se comprueba experimentalmente en cada caso.

En este apartado se verán los conceptos generales y las propiedades de los tipos de segmentación más comunes.

**Segmentación basada en características de píxel.** Se evalúa cada píxel en función de las características locales de la imagen en el píxel (y usualmente también sus vecinos), y se decide a qué región (también conocido como segmento) pertenece. Este tipo de segmentación se usa comúnmente cuando se requiere separar objetos con similares características de color o intensidad de un fondo heterogéneo. El caso ideal es aquel en el cual los objetos poseen un rango de colores o intensidad de gris muy estrecho, siendo el fondo uniforme. En tal caso se puede definir un umbral de segmentación para separar objeto del fondo. A esta técnica de asignación de un umbral se la conoce como *thresholding* (literalmente "umbralización").

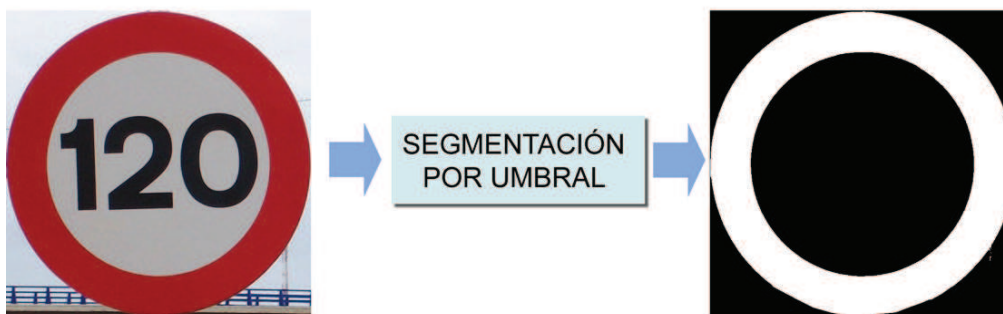


Figura 3.26. Segmentación basada en umbral de intensidad de gris.

En el *thresholding* se define un valor umbral y se toman los píxeles en este rango según pertenezcan o no al fondo: se toman los que no pertenecen al fondo y se rechazan todos los demás. Una imagen de este tipo se muestra como una imagen binaria (de dos niveles) utilizando blanco y negro u otros colores para distinguir las regiones (no hay una convención estándar sobre cuáles son los rasgos de interés, si los blancos o los negros, así que la elección varía en cada caso). [41].

Suponiendo que el histograma de nivel de gris de la **Figura 3.26** corresponde a una imagen  $f(x,y)$ , compuesta por objetos oscuros sobre un fondo brillante de tal forma que los píxeles de objetos y fondo son modos de selección, una forma obvia de extraer los objetos del fondo es seleccionar un umbral  $T$  que separe estos modos; después, cualquier punto  $(x,y)$  para el que  $f(x,y) > T$  se denomina un punto del objeto; cualquier otro punto, se denomina punto del fondo.

Algunas ventajas de la segmentación basada en píxel son:

- El uso de la segmentación por color elimina los colores indeseados, y por ende el número de bordes de la imagen se decrementa, lo cual resulta útil como etapa previa a una detección de bordes. En tal caso, la complejidad computacional de un detector de bordes disminuye.
- Con la ayuda de la segmentación por color, el número de detecciones fallidas se decrementa en una posterior etapa de detección de formas. Esto resulta útil para la detección de objetos que tienen colores y formas muy definidas, como las señales de tráfico.

**Segmentación basada en modelos. Transformada de Hough.** La transformada de Hough es una herramienta que permite detectar curvas en una imagen. Se basa en la búsqueda de características geométricas de los objetos: rectas, triángulos, objetos circulares, etc. La transformada de Hough es una de las técnicas de segmentación basada en modelos más

utilizadas, debido a su robustez frente al ruido y a su comportamiento ante la existencia de huecos en la frontera del objeto. A la hora de aplicar la transformada de Hough a una imagen es necesario obtener primero una imagen binaria de los píxeles que forman parte de la frontera del objeto usando, por ejemplo, segmentación basada en umbral. El objetivo de la transformada de Hough es encontrar puntos en la imagen que estén alineados. Esto se reduce a hallar los píxeles de una imagen que satisfagan la siguiente ecuación de la recta en coordenadas polares, para distintos valores de  $\rho$  y  $\theta$ :

$$\rho = x \cdot \cos \theta + y \cdot \text{sen} \theta$$

Por lo tanto, la transformada de Hough requiere una transformación del espacio de coordenadas  $(x,y)$  en el espacio polar de parámetros  $(\rho, \theta)$ . En esta transformación, una recta en el espacio  $(x, y)$  que esté a distancia  $\rho_j$  del origen y posea pendiente  $\theta_i$ , se representa como un sólo punto  $(\rho_j, \theta_i)$  en el espacio transformado (**Figura 3.27**).

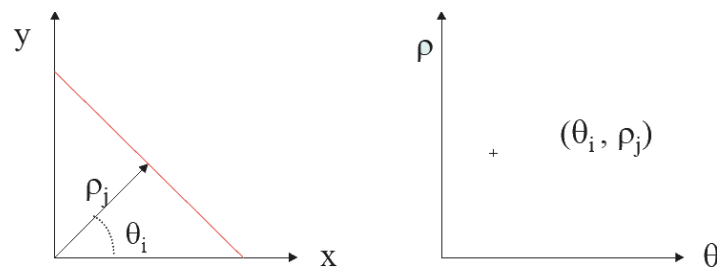


Figura 3.27. Transformada de Hough de una recta.

Así mismo, la transformada de un punto en el plano  $(x,y)$  corresponde a una curva sinusoidal en el plano  $(\rho, \theta)$  (**Figura 3.28**). Es importante destacar que los puntos de cruce de todas las curvas en el espacio de Hough, definen la recta a la que pertenecen dichos puntos en el espacio imagen.

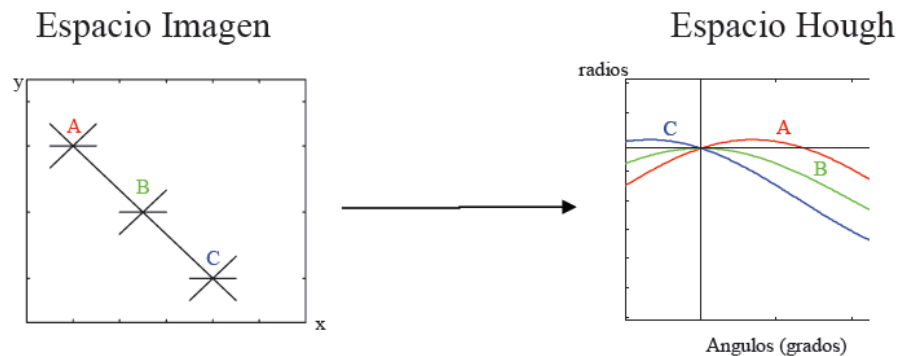


Figura 3.28. Transformada de Hough de tres puntos A, B, C.

A partir de la Transformada de Hough, es posible seleccionar píxeles que pertenezcan a rectas de interés. Para ello, se toma como característica discriminadora los valores de  $(\rho_j, \theta_i)$  deseados, y para cada píxel de la imagen original, se calcula el  $\rho$  y el  $\theta$  correspondiente, manteniendo el píxel en uno y otro segmento según cumpla con los valores discriminatorios.



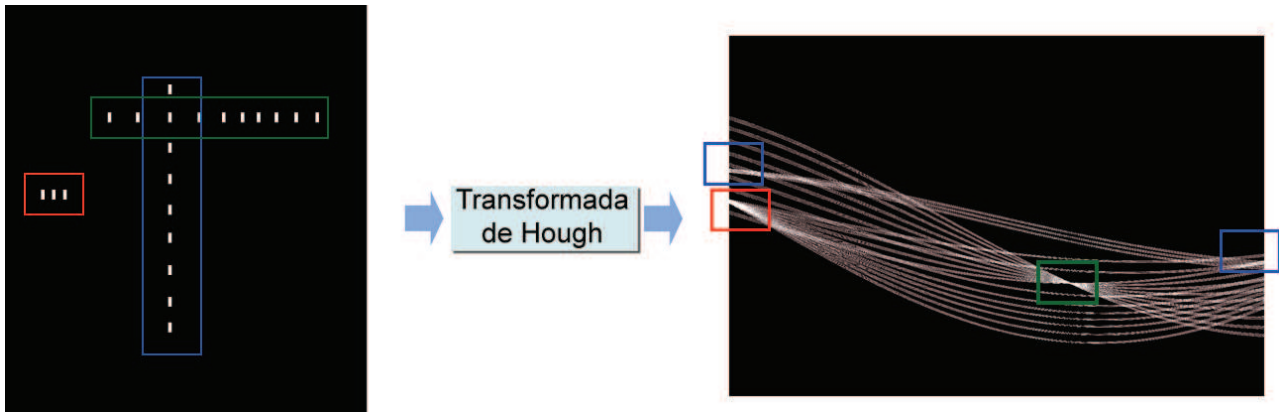


Figura 3.29. Ejemplo de la Transformada de Hough.

La Transformada de Hough inicialmente se aplicó para la detección de rectas sobre imágenes, aunque más tarde se extendió para ser usada con cualquier tipo de curva que pudiera describirse de forma paramétrica (triángulos, círculos, elipses, rectángulos, etc.), conociéndose el método como Transformada de Hough Generalizada.



Figura 3.30. Aplicación de la Transformada de Hough Generalizada en un partido de Fútbol [53]

Finalmente, a modo de conclusión se exponen las ventajas e inconvenientes de la aplicación de esta técnica. Como ventajas se pueden señalar:

- Cada píxel de la imagen se procesa de modo independiente, lo que facilita su implementación en paralelo.
- La transformada general de Hough es útil para la detección de formas complejas.
- Es capaz de reconocer patrones ligeramente deformados, ocultos o discontinuos.
- Robusta frente al ruido.
- Permite buscar simultáneamente todas las ocurrencias de un patrón.

En cuanto a los inconvenientes, los más importantes son:

- El tiempo de computación y memoria usados es alto. La aplicación de la Transformada de Hough consume muchos recursos.
- No ofrece respuesta absoluta, sino un índice de probabilidad de que cada una de las formas posibles sea la buscada.

### 3.2.3.2 Etiquetado de componentes conectados

El etiquetado de componentes conectados, o simplemente etiquetado, es una operación que agrupa los píxeles correspondientes al mismo objeto y les asigna una etiqueta, separando así unos objetos de otros. Este proceso se realiza usualmente cuando la imagen ha sido binarizada previamente (Por ejemplo tras una segmentación por umbral como la de la **Figura 3.26**). Como resultado, se obtiene una imagen en la cual se separa cada objeto con una etiqueta diferente, pudiendo entonces extraer características de los mismos, como su centroide, sus coordenadas o su tamaño, o conocer el número de objetos en una imagen [54].

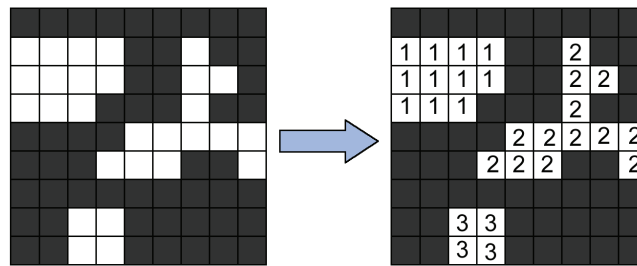


Figura 3.31. Imagen original y resultado del etiquetado.

Debido a la naturaleza del escaneo de la imagen (de arriba a abajo y de izquierda a derecha), los algoritmos para etiquetar dan buenos resultados con objetos convexos, pero presentan problemas cuando aparecen objetos que tienen concavidades (formas en U), como se observa en la **Figura 3.32**. En la figura se observa que diferentes partes de un mismo objeto pueden etiquetarse con diferentes valores, llegando un punto en la imagen donde se produce una colisión de etiquetas. En este sentido el peor caso que puede plantearse es un objeto con forma de espiral [55]. Como se aprecia en la **Figura 3.32**, en primer lugar se encuentra un píxel (p1) sin etiquetar, y se le asigna una etiqueta nueva (color rojo). Más adelante, se encuentra un nuevo píxel que aparentemente pertenece a un nuevo objeto, y se le asigna una etiqueta nueva (color azul). Más tarde en la imagen, se "descubre" que los objetos que en un principio parecían diferentes, son en realidad un mismo objeto. En este caso se detecta una colisión, en donde se llega a la conclusión de que las etiquetas rojas y azules pertenecen en realidad al mismo objeto.

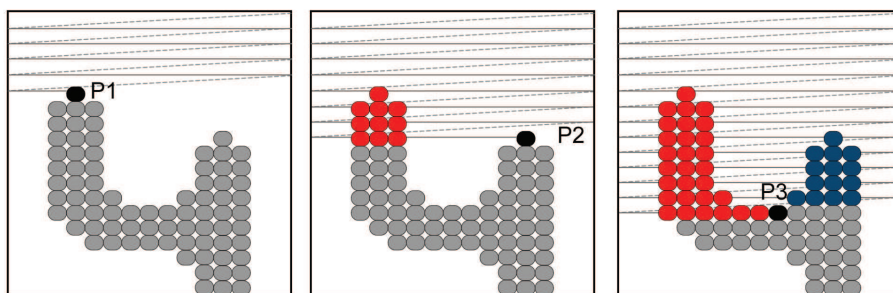


Figura 3.32. Colisión de etiquetas en objetos cóncavos.

Las colisiones de etiquetas deben ser tratadas con mucho cuidado, y en general se utiliza el método de fusión de etiquetas. Este consiste en que al detectar una colisión, se decide cual es la etiqueta menor (por ejemplo las de color rojo), y todos los píxeles etiquetados en azul cambian su valor de etiqueta a rojo. Éste procedimiento, a pesar de ser un método comúnmente usado en imágenes almacenadas en una memoria, es inviable en sistemas de vídeo, sistemas basados en FPGA o sistemas en tiempo real, ya que las imágenes fluyen sin almacenarse y por lo tanto no es posible acceder a cualquier píxel de la imagen en cualquier momento para cambiar su etiqueta. Es por ello que existen ciertas técnicas y algoritmos que se verán en este apartado.

Para los sistemas descritos anteriormente, en los cuales la imagen se escanea de arriba a abajo y de izquierda a derecha, la máscara utilizada es la siguiente:

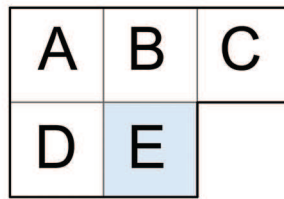


Figura 3.33. Máscara para el etiquetado de componentes conectados.

En general, el algoritmo para la detección de componentes conectados es el siguiente:

- Si  $E = '0'$ , entonces se asigna la etiqueta de fondo al píxel actual.
- Si  $A, B, C, D = '0'$  (píxeles de fondo) y  $E = '1'$ , entonces se asigna una nueva etiqueta al píxel actual.
- Si los vecinos  $A, B, C, D$ , distintos de cero son iguales y  $E = '1'$ , entonces la etiqueta asignada al píxel actual será la etiqueta común a los vecinos.
- Si los vecinos  $A, B, C, D$ , poseen diferentes etiquetas y  $E = '1'$ , entonces la etiqueta asignada al píxel actual será la menor de todas. En este caso se deberá proceder a la fusión de las etiquetas mayores con la menor.

Según el sistema utilizado, los recursos y el tiempo disponible que se necesiten para realizar esta técnica, existen distintos tipos de algoritmos, que se resumen a continuación.

**Algoritmo clásico o de dos pasadas.** Una de las primeras publicaciones que describía este algoritmo fue escrita por Rosenfeld y Pflatz [56]. El algoritmo de dos pasadas es comúnmente denominado como "clásico", y su característica clave es el número constante de pasadas (dos pasadas) a través de la imagen binaria para realizar el etiquetado. La mayoría de sistemas que utilizan etiquetado de componentes conectados, utilizan este algoritmo, aunque se varía a menudo la forma de administrar los datos y las tablas de equivalencias. El Algoritmo clásico consiste en dar una primera pasada sobre la imagen binaria, asignando etiquetas preliminares. Cuando se encuentren colisiones, se actualizarán los datos en la tabla de equivalencia indicando qué etiquetas pertenecen al mismo objeto. Al final del primer escaneo, la tabla de equivalencias es ordenada de menor a menor, y en una segunda pasada se sobrescriben todas las etiquetas mayores que han colisionado con una menor.

El principal inconveniente del algoritmo clásico es el consumo de memoria de la imagen etiquetada de salida, ya que la imagen etiquetada que se obtiene tras la primera pasada debe almacenarse en memoria antes de comenzar el segundo escaneo, y el número de etiquetas utilizadas en el primer escaneo es muy dependiente de la complejidad de la imagen.

**Algoritmo de múltiples escaneos.** Este algoritmo fue propuesto en 1981 por Haralick [57], y cuenta con la ventaja de no necesitar de una memoria para almacenar las equivalencias ocurridas durante los pases. Esta técnica implica múltiples pasadas sobre la imagen binaria, tanto hacia atrás como hacia adelante, hasta que no ocurra ningún cambio de etiquetas. Todas las colisiones de etiquetas son resueltas en el contexto de los píxeles vecinos. Este sistema fue propuesto para sistemas con limitaciones de los recursos de memoria, y para imágenes de baja resolución, y no se recomienda para imágenes de alta resolución.

**Algoritmo de procesamiento paralelo.** Este algoritmo fue creado en un principio para plataformas de procesamiento en paralelo, y no se aplica en arquitecturas de computadores ordinarias. Sin embargo, este tipo de algoritmos, aunque son realizables en FPGA, requieren

grandes cantidades de recursos para llevarse a cabo, y en la actualidad no son eficientes para el streaming de vídeo, ni las imágenes de alta resolución.

**Algoritmo por seguimiento de contorno.** Fue introducido en 2003 por F. Chang y J. Chen, [58] y usa ciertas técnicas de detección de contornos para detectar los objetos, y posteriormente rellenar el resto de píxeles interiores con las etiquetas correspondientes. Este método tiene la ventaja de necesitar solamente una pasada para etiquetar todos los contornos, utilizando menos recursos y memoria que los algoritmos basados en tabla de equivalencias. Tampoco tiene sentido hablar de colisión de etiquetas, ya que la imagen es escaneada una vez. Sin embargo, este algoritmo requiere acceso aleatorio a todos los píxeles de la imagen, por lo cual se convierte en un algoritmo no implementable en sistemas en tiempo real, o streaming de vídeo, ya que se necesita que la imagen a analizar esté almacenada en una memoria.

**Algoritmo de pase simple.** Este tipo de algoritmos es relativamente nuevo [59], y fue creado específicamente para etiquetado de componentes conectados en sistemas de streaming de vídeo y sistemas en tiempo real. El etiquetado se realiza en una sola pasada, mientras la imagen va llegando en streaming, de arriba a abajo y de izquierda a derecha. La ventaja más significativa de este algoritmo es que no se necesita almacenar todas las etiquetas de un fotograma completo, sino que todo se hace en el contexto de las vecindades del píxel. Otra de sus características más novedosas, es que al mismo tiempo que se etiquetan los píxeles de la imagen binaria, se van extrayendo las características de los objetos: su tamaño, su número, su centro, su posición en la imagen, etc.. Se mantiene una tabla de equivalencias, donde se resuelven las colisiones de etiquetas, y también una tabla de características, donde se van añadiendo las coordenadas.

En el **Apartado 6.15.3** se verá en detalle cómo implementar un algoritmo de pase simple en una FPGA.

### 3.3 Vídeo digital

En este apartado se pretende introducir las particularidades del análisis y procesamiento de imágenes aplicadas al vídeo digital. En apartados anteriores ya se hacía referencia a la naturaleza de los datos cuando se trata de un sistema de vídeo; en particular, los más importantes son los siguientes:

- Los datos fluyen a través de un bus y para acceder a un fotograma completo, se hace uso de un *Frame Buffer*.
- Existe un tiempo limitado para realizar las operaciones requeridas en cada fotograma, y este viene dado por la cantidad de fotogramas por segundo del vídeo.
- Cada fotograma del vídeo cuenta con un espacio de *blanking*, además del espacio activo, que podrá ser utilizado para realizar tareas "entre fotogramas".
- El flujo de datos en vídeo es muy alto, sobre todo en vídeo de alta resolución, por lo cual los sistemas basados en FPGA con procesamiento paralelo son muy adecuados, ofreciendo buenos resultados en tiempo real.

#### 3.3.1 Video Timing y tasa de datos

En la **Figura 3.34** se puede observar la composición típica de un fotograma de vídeo digital. Los espacios que se encuentran a la izquierda y en la parte superior del fotograma, son los llamados espacios de *blanking*, y en ellos no se muestra información por la pantalla.

Originalmente, una imagen de vídeo analógica se formaba con la sucesión de un número determinado de imágenes por segundo en la pantalla. Cada una de estas imágenes a su vez componía de un número determinado de líneas que cubren la pantalla de forma horizontal. Los

principales sistemas de video analógico eran PAL, que disponía de 625 líneas por cuadro y 25 f.p.s. (fotogramas por segundo) y NTSC con 525 líneas por fotograma y 30 f.p.s. Estas líneas eran mostradas gracias a una pantalla de tubo de rayos catódicos (CRT), que disponía de un haz de electrones controlado por bobinas magnéticas que excitaban una pantalla de vidrio recubierta de fósforo y plomo.

Para dar tiempo suficiente a las bobinas magnéticas a dirigir el rayo a la parte izquierda en cada cambio de línea, y a la parte superior al finalizar un fotograma, se crearon los tiempos de *blanking*, en los cuales el haz de electrones volvía a la posición deseada.

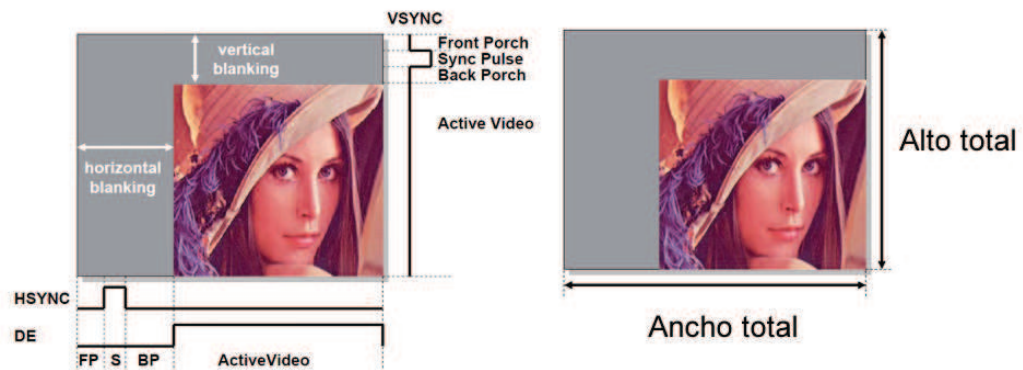


Figura 3.34. Video timing y tamaño total de un fotograma.

A pesar de que hoy en día la naturaleza del vídeo y de los monitores ha cambiado, se siguen conservando estos espacios de "pausa" entre fotogramas, así como las señales Vsync / Hsync. Estos espacios son utilizados para diversas tareas, como el transporte de datos de broadcast, copyright, procesamiento de imágenes, actualización de buffer, teletexto, entre otros.

Cabe destacar que para el cómputo de la tasa de píxel de un vídeo y el flujo de datos que genera, se tiene en cuenta tanto el tamaño del fotograma, como los espacios de *blanking*, hecho que puede observarse en la siguiente tabla, en la que se ha tomado como referencia un sistema de vídeo digital XGA con resolución de 1024x768 píxeles.

- **Vídeo Activo:**
  - Ancho: 1024 píxeles
  - Alto: 768 píxeles
- **Vídeo Activo con espacios de *blanking*:**
  - Ancho: 1334 píxeles
  - Alto: 806 píxeles
- **Tasa de Píxel:**
  - $(1344 \times 806) \times (60 \text{ Fps}) = 65 \text{ MHz}$
- **Tamaño de Almacenamiento:**
  - 24 bits per pixel : R[7:0], G[7:0], B[7:0]
  - $(1024 \times 768) \times (3 \text{ Bytes/píxel}) = 2.25 \text{ MB / frame}$
- **Tasas de Datos:**
  - $(1024 \times 768) \times (60 \text{ Fps}) \times (24 \text{ bits/píxel}) = 1.13 \text{ Gbps (sólo vídeo)}$
  - $(1344 \times 806) \times (60 \text{ Fps}) \times (24 \text{ bits/píxel}) = 1.56 \text{ Gbps (vídeo + blanking)}$

En la **Tabla 3.3** se muestran las distintas tasas para los modos de vídeo más comunes en la actualidad [60].

Video Resolution	Image Size	Pixel Rate	Pixel Size	Storage Size	Data Rate
NTSC (60 Hz)	720x480	13.5 MHz	16 bits	165 KB	75 Mbps
PAL (50 Hz)	720x576	13.5 MHz	16 bits	198 KB	75 Mbps
VGA (60 Hz)	640x480	25.175 MHz	24 bits	879 KB	441 Mbps
SVGA (60 Hz)	800x600	40 MHz	24 bits	1.373 MB	695 Mbps
XGA (60 Hz)	1024x768	65 MHz	24 bits	2.250 MB	1.13 Gbps
HD 720p (60 Hz)	1280x720	74.25 MHz	24 bits	2.637 MB	1.3 Gbps
SXGA (60 Hz)	1280x1024	110.0 MHz	24 bits	3.750 MB	1.9 Gbps
HD 1080p (60 Hz)	1920x1080	148.5 MHz	24 bits	5.933 MB	3.0 Gbps
UXGA	1600x1200	162.0 MHz	24 bits	5.493 MB	2.9 Gbps
HDMI 1.3	2560x1600 1920x1200	340.0 MHz 340.0 MHz	24 bits 48 bits	11.719 MB 13.184 MB	8.2 Gbps 16.4 Gbps

Tabla 3.3. Tasas de datos para distintas resoluciones de vídeo digital (sólo vídeo) [60].

### 3.3.2 Conceptos básicos de procesamiento de vídeo

La dimensión temporal que añade el vídeo digital abre las puertas a nuevas formas de procesamiento. Éstas pueden englobarse en tres grandes grupos: procesado individual o píxel a píxel, procesado espacial (similar a los métodos vistos en el procesamiento de imágenes, pero aplicado a cada fotograma del vídeo) y por último el procesado temporal. A continuación se hará una breve descripción de los mismos.

#### 3.3.2.1 Procesado píxel a píxel

Este método recibe como entrada un píxel de una o varias fuentes, y saca por la salida un píxel resultado. Algunos ejemplos son:

- **Alpha Blending.** Sirve para fundir fotogramas de dos fuentes de vídeo de entrada. Consiste en multiplicar cada los píxeles de las entradas por una constante (no necesariamente la misma) y sumarlos en un píxel de salida.
- **Color Correction.** Las componentes RGB del píxel de salida serán una combinación lineal de las componentes RGB del píxel de entrada.
- **Gamma.** La intensidad del píxel de salida viene dada por una operación no lineal de la intensidad del píxel de entrada.

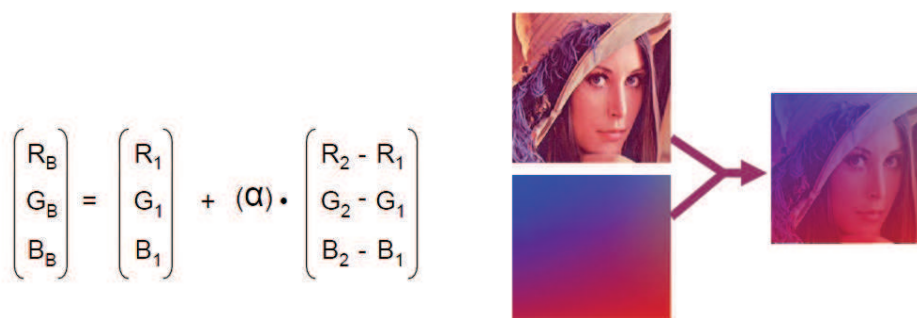


Figura 3.35. Ejemplo de Alpha Blending sobre dos fuentes de vídeo.

### 3.3.2.2 Procesado espacial

En el procesado espacial de vídeo se aplican todos los conocimientos y métodos del análisis y procesamiento de imágenes visto en apartados anteriores, con la restricción añadida del tiempo de procesamiento, que queda acotado por la tasa de píxeles del vídeo. En sistemas de tiempo real, esta tasa de píxeles resulta de vital importancia ya que todas las operaciones necesarias deberán hacerse en el tiempo de un sólo fotograma.

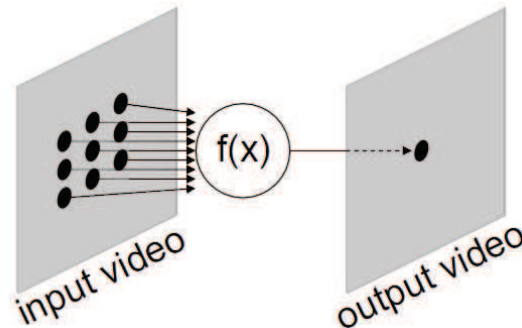


Figura 3.36. Procesamiento espacial de vídeo [60].

Algunos ejemplos de procesado espacial son el escalado, los filtros 2D o el "demosaijing".

### 3.3.2.2 Procesado temporal

El procesado temporal recibe como entrada  $N$  píxeles y saca como resultado un píxel de salida, con la particularidad de que los  $N$  píxeles de entrada no pertenecen al mismo fotograma. Se utiliza para la compresión de datos de vídeo, supresión de redundancia, la reconstrucción de imágenes parciales, mejora de la resolución, la detección de movimiento, compensación del movimiento, desentrelazado, aplicación de filtros temporales, entre otros.

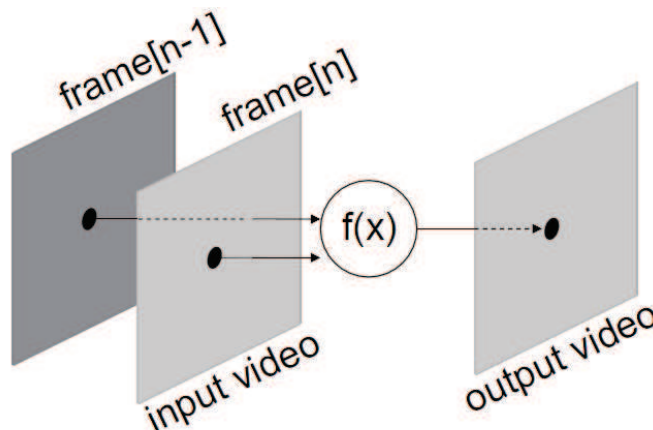


Figura 3.37. Procesamiento temporal de vídeo [60].

El procesado temporal requiere que el sistema almacene al menos  $N-1$  fotogramas en un *frame buffer*, con el consiguiente gasto computacional y de memoria. Así mismo, se requiere de una memoria multi-puerto, capaz de leer y escribir datos simultáneamente en el mismo espacio de memoria.