

**UNIVERSIDAD POLITECNICA DE VALENCIA**  
**ESCUELA POLITECNICA SUPERIOR DE GANDIA**  
**I.T. TELECOMUNICACIÓN (IMAGEN Y SONIDO)**

---



UNIVERSIDAD  
POLITECNICA  
DE VALENCIA



ESCUELA POLITECNICA  
SUPERIOR DE GANDIA

**“Desarrollo de una interfaz gráfica  
en MatLab para la aplicación de  
detección de humos mediante  
procesamiento digital de  
imágenes.”**

**TRABAJO FINAL DE CARRERA**

Autora:  
**Alba María Mancebo Pérez**

Tutor/a:  
**Ignacio Bosch Roig**  
**Jose Ignacio Herranz Herruzo**

**GANDIA, 2012**



# Índice General:

Antecedentes del proyecto y objetivo.....	Página 6
---	----------

## PARTE 1:

### 1. Introducción

1.1. Fundamentos.....	Página 8
1.2. Representación de imágenes.....	Página 12
1.3. Etapas fundamentales del procedimiento.....	Página 14
1.4. Elementos de los sistemas de procesamiento digital de imágenes.....	Página 15

## PARTE 2:

### 2. Tratamiento Digital de la imagen

2.1. Fundamentos de la imagen digital.....	Página 17
2.1.1. Percepción visual.....	Página 17
2.1.2. Muestreo y cuantificación.....	Página 18
2.1.3. Relaciones básicas entre píxeles.....	Página 19
2.1.4. Geometría de la imagen.....	Página 21
2.2. Mejora de la imagen.....	Página 23
2.2.1. Procesado de punto.....	Página 23
2.2.1.1. Transformaciones de intensidad	
2.2.1.2. Procesado del histograma	
2.2.1.3. Substracción de imágenes	
2.2.1.4. Media de imágenes	
2.2.2. Filtrado espacial.....	Página 27
2.2.2.1. Filtros de suavizado	
2.2.2.2. Filtros de realzado	
2.2.3. Dominio de la frecuencia.....	Página 29
2.3. Restauración de imágenes.....	Página 32
2.4. Segmentación.....	Página 34
	Página 34

2.4.1. Segmentación por umbralización.....	Página 36
2.4.2. Segmentación por detección de regiones.....	Página 36
2.4.3. Segmentación por detección de fronteras.....	Página 38
<b>2.5. Representación y descripción.....</b>	<b>Página 40</b>
2.5.1. Descriptores de frontera.....	Página 40
2.5.2. Descriptores de región.....	Página 43
2.5.3. Descriptores relacionales.....	Página 44
<b>2.6. Reconocimiento e interpretación.....</b>	<b>Página 46</b>
2.6.1. Métodos de decisión.....	Página 46
2.6.1.1. Métodos Matching	
2.6.1.2. Redes neuronales	
2.6.2. Métodos estructurales.....	Página 49
<b>PARTE 3:</b>	
<b>3. Implementación de algoritmos para el procesado.....</b>	<b>Página 53</b>
<b>3.1. Introducción a la herramienta de trabajo</b>	
<b>GUIDE.....</b>	<b>Página 54</b>
3.1.1. Manejo de datos entre los elementos de la aplicación y el archivo .m.....	Página 54
3.1.2. Sentencias GET y SET.....	Página 55
3.1.3. Iniciando GUIDE.....	Página 55
3.1.4. Partes de GUIDE.....	Página 56
3.2. Entorno gráfico de trabajo.....	Página 58
3.3. Algoritmo para el procesado.....	Página 62
<b>4. Conclusiones y líneas de trabajo futuro.....</b>	<b>Página 65</b>
<b>5. Bibliografía.....</b>	<b>Página 66</b>
<b>6. Índice de figuras.....</b>	<b>Página 67</b>



## **Antecedentes del proyecto:**

Este proyecto esta basado en el proyecto de fin de carrera de Rubén Llobregat Rubio titulado: “Procesado digital de imágenes de video para la detección de humo”.

Con la ayuda de mi profesor Ignacio Bosh Roig, hemos intentado mejorar dicho programa, añadiéndole varias ventanas a la ventana principal para lograr entender mejor el procesado que estamos realizando entre otras cosas.

## **Objetivo principal:**

Realización de un programa capaz de detectar el humo en un video dónde la cámara permanezca fija. Para ello utilizaremos tratamiento digital de cada frame en el programa MatLab con la interfaz GUI hasta lograr la detección deseada.

---

# Parte 1:

# Introducción

---

# 1. INTRODUCCIÓN

## 1.1. FUNDAMENTOS

El procesamiento digital de imágenes es un tema en constante evolución, ya sea por motivos necesarios como la medicina o la misma inquietud humana. Es un campo muy abierto a todos los ámbitos y con infinitud de aplicaciones, como la nombrada anteriormente de la medicina u otras como la astronomía, geología, microscopía, etc.

La percepción psicovisual del ser humano es un aspecto del tratamiento digital de imágenes muy importante a tener en cuenta ya que a parte de las distintas técnicas que empleemos a la hora de tratar una imagen el resultado final tiene que ser aprobado por el ser humano, y ahí es donde entra su percepción.

El tratamiento digital de imágenes lo utilizamos en la detección de movimiento para para ver las diferencias entre imágenes de un misma escena tomadas en distintos instantes de tiempo. Puede haber cambios de movilidad, luminosidad y objetos que cambian de forma con el tiempo. Como tratamiento digital de imágenes podemos considerar cualquier ajuste, mejora o modificación de la imagen.

Un resumen de las aplicaciones relacionadas con el tratamiento digital de imágenes podría ser:

- Mejora de las imágenes para interpretación humana, hacer la información relevante más visible:

**Realzado:** contraste, nitidez, ecualización, pseudocolores

Para comprender realmente el realzado de imágenes es necesaria la noción previa de conceptos como los bordes, que los podríamos definir como las líneas que separan las regiones de una imagen con diferentes tonalidades de grises. Por otro lado tenemos el concepto de contraste de una imagen que viene dado por la repartición de los diferentes tonos de gris de cada pixel de una imagen. Con ello podemos decir que el realzado de una imagen consiste en realzar la información más relevante, realzar los bordes de una imagen y hacer que ésta sea más nítida; aun así el concepto de realzado es algo ambiguo ya que no existen imágenes buenas y malas claramente, depende de la percepción de cada uno. Existen diferentes técnicas de realzado que explicaremos más adelante; y la combinación de éstas junto con la percepción del interesado logrará el resultado de una buena imagen. Con el realzado de imágenes procesamos la imagen para conseguir un resultado útil para una aplicación en concreto.

**Restauración:** desenfocado, movido, interferencias, perspectiva

Con restauración nos referimos a mejorar o reconstruir una imagen que ha sido degradada por algún fenómeno, ya sea provocado por el paso del tiempo como es el



caso de las fotos antiguas; o por defectos provocados en la toma de ésta, como movimientos, interferencias, perspectiva, etc.

Las técnicas de restauración han mejorado mucho en los últimos años, antes por ejemplo. Se basaban en métodos en el dominio de la frecuencia, y actualmente se basan más en métodos algebraicos y manipulación de grades sistemas de ecuaciones.

En la figura que podemos ver a continuación podemos ver un ejemplo de mejora de imágenes para la interpretación humana.

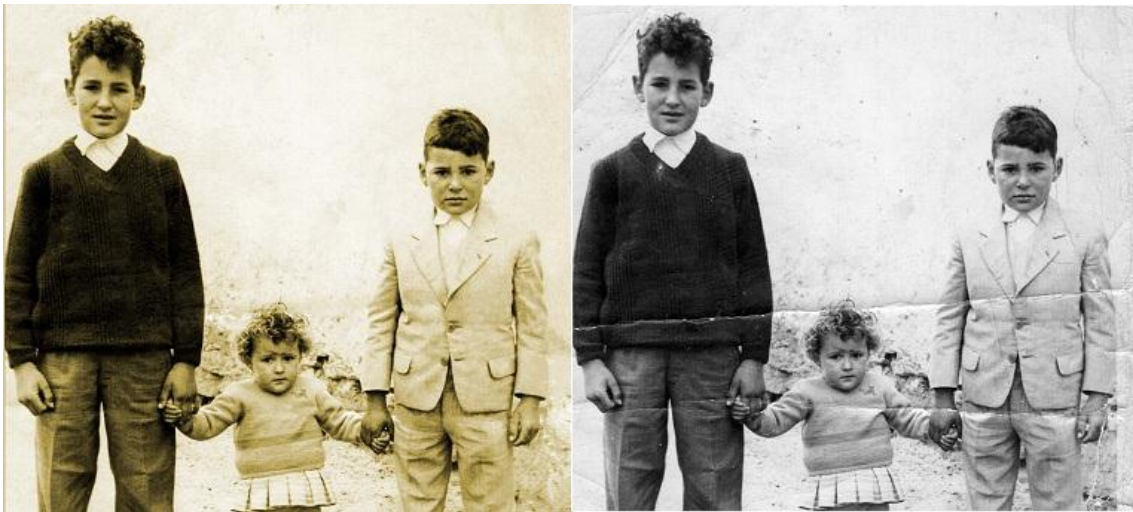


Figura 1.1.1. Restauración de Imágenes.

- Tratamiento de escenas para percepción automática por ordenador (Visión artificial)

Como visión artificial entendemos la detección, segmentación, localización y reconocimiento de ciertos objetos en imágenes de forma automática a través del ordenador. Se pretende que el ordenador entienda una determinada escena y a partir de ella extraiga la información que ara el caso sea relevante.

En este ámbito existen infinidad de aplicaciones como: robótica industrial, OCR's (reconocimiento óptico de caracteres), reconocimiento automático de blancos, cuenta de microorganismos, cuenta de objetos y personas, meteorología...

En la figura inferior se muestra un ejemplo de una red neuronal utilizada en un sistema OCR para el reconocimiento de caracteres.

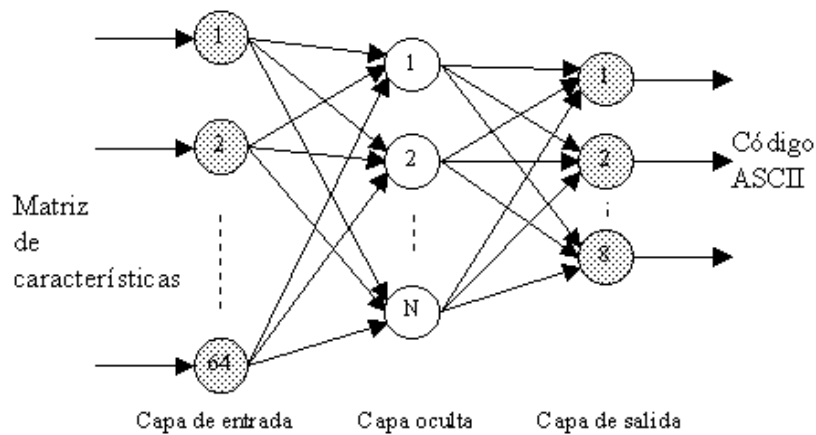


Figura 1.1.2. Estructura de una red neuronal de reconocimiento de caracteres.

- Compresión ya sea para almacenamiento o para transmisión.

Uno de los principales inconvenientes de las imágenes digitales es su tamaño, por lo que nos vemos obligados a emplear técnicas de compresión. Aunque las capacidades de almacenamiento evolucionan rápidamente y cada vez son mayores, también nos vamos acostumbrando a trabajar con más pixels por lo que las técnicas de compresión tampoco dejan sus investigaciones.



Figura 1.1.3. Imagen original e imagen comprimida.

Una breve introducción de las técnicas para llevar a cabo una detección de movimiento podría ser la siguiente:

- Métodos de pre-procesado

Dentro de estos métodos se explicará lo anteriormente nombrado como realizado de imágenes, dónde se mejorará la imagen para la interpretación del ojo humano dedicado a una determinada aplicación.

- Diferenciación

Se trata de una técnica donde se realiza la resta de dos imágenes para ver los píxeles diferentes entre ambas tomadas de una misma escena en distintos instantes de tiempo. Es una técnica básica para la detección de movimiento.

- Histograma

El histograma es una representación gráfica en forma de barras donde se indica el número de píxeles que tienen un determinado nivel de gris. Cada barra simboliza un nivel de gris distinto, y la altura de esta barra la cantidad de píxeles de la imagen que tienen ese nivel de gris.

- Segmentación

Mediante la segmentación dividiremos la imagen en partes u objetos que la forman. Es uno de los procesos más importantes y difíciles de realizar de forma automática, ya que este determinará el éxito o fracaso del proceso de análisis.

Tenemos varios tipos de segmentación que más tarde explicaremos:

- Por umbralización: simple, multinivel, bidimensional, multibanda, por textura...
- Por detección de discontinuidades: de puntos, de líneas o de bordes.
- Orientada a regiones.

- Procesado morfológico

Este proceso es un proceso matemático que se encarga de extraer componentes de una imagen que son útiles en la representación y descripción de objetos. Como por ejemplo límites o bordes, esqueletos, etc.

- Representación del color

Los espacios de color utilizados tales como: cosificación RGB (Rojo, Verde y Azul), HSL (Matiz, Saturación, Luminancia), CMYK (Cyan, Magenta, Amarillo, Negro), YUV o YIQ.

## 1.2. REPRESENTACIÓN DE IMÁGENES

Una imagen digital a niveles de gris es una función bidimensional, una función 2D, cuyos valores se han obtenido muestreando tanto en coordenadas espaciales como en luminosidad; y puede ser considerada como una matriz cuyos valores de cada elemento representan el nivel de gris de cada punto. Los elementos de dicha matriz son denominados puntos o píxeles.

Llamaremos cuantificación a la discretización de la amplitud y muestreo a además de la discretización de las coordenadas espaciales.

Podemos considerar que una imagen será más buena cuántos más niveles de gris tengamos para representar la imagen y mayor sea el número de puntos, ya que obtendremos mayor nivel de detalle.

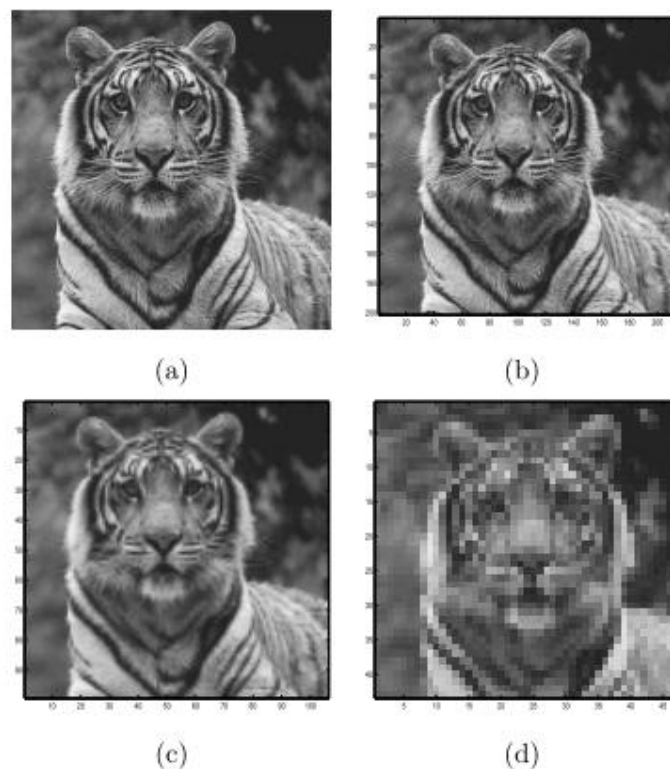


Figura 1.2.1. (a) Imagen original; (b) Imagen muestreada cada 2 píxeles;  
(c) Imagen muestreada cada 4 píxeles; (d) Imagen muestreada cada 8 píxeles

En la figura 1.2.1 podemos observar una imagen muestreada con diferentes valores, y vemos que cuanto más disminuimos la frecuencia de muestreo de la imagen

pero resolución tendremos de la imagen. También cabe la posibilidad de que utilicemos pocos niveles de gris para representar la imagen y aparezcan falsos contornos.

En el tratamiento de imágenes se pueden destacar tres etapas principales:

1. Adquisición de la imagen.
2. Procesamiento de la imagen.
3. Presentación al observador.

Todos los conceptos vistos hasta ahora los desarrollaremos más adelante con más exactitud.

### 1.3. ETAPAS DEL PROCEDIMIENTO

Las fases del procesado de imágenes las podríamos esquematizar de la siguiente manera:

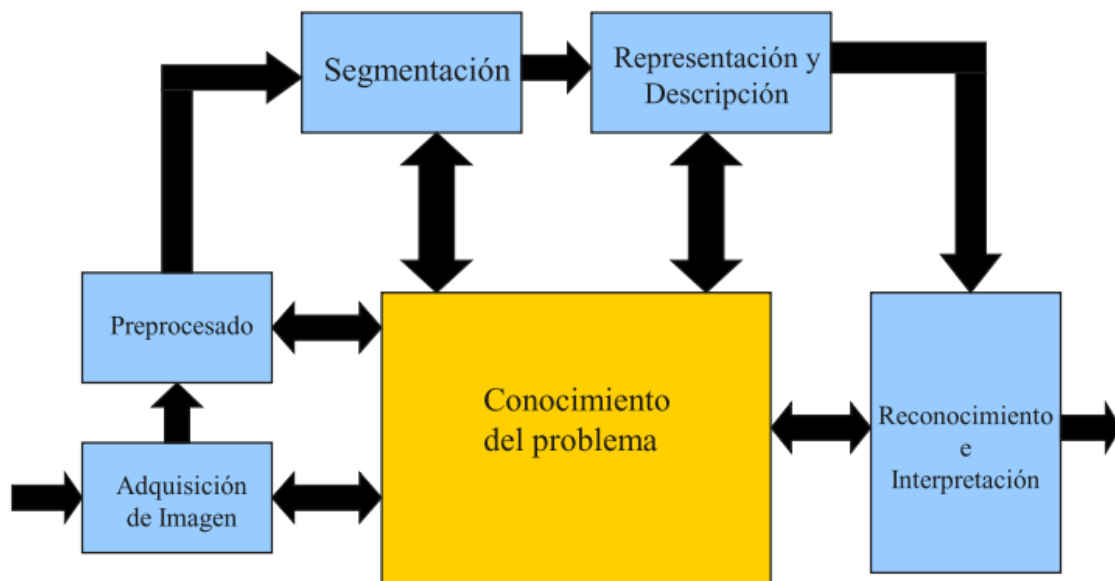


Figura 3.1.1 esquema de procesado digital de imágenes.

1. Adquisición de la imagen: mediante cámara de televisión, cámara fotográfica y/o digital, escáneres, tarjetas digitalizadoras o bases de datos.
2. Procesado: conjunto de técnicas que se aplican a las imágenes digitales con el objetivo de mejorar la calidad o facilitar la búsqueda de información.
3. Segmentación: se trata de dividir una imagen digital en varias partes u objetos.
4. Representación y descripción: identificar la imagen, describirla a través de la medida de parámetros de la misma.
5. Reconocimiento e Interpretación: última fase del procesado de la imagen donde tratamos que una máquina adquiera la habilidad humana de la visión.

Podemos agrupar como procesado a nivel bajo la adquisición de la imagen y el procesado, procesado a nivel intermedio, segmentación y representación y descripción; y por último, el reconocimiento e interpretación como procesado a alto nivel.

#### **1.4. ELEMENTOS DE LOS SISTEMAS DE PROCESAMIENTO DIGITAL DE IMÁGENES.**

1. Adquisición de imágenes: se necesitan dos elementos básicamente, un elemento físico sensible a una determinada banda del espectro de energía electromagnética y que produzca una señal eléctrica de salida proporcional al nivel de energía detectado; y otro que convierta la señal de salida a forma digital, digitalizador.
2. Almacenamiento: elementos tales como: CD, DVD, pen drive, Cinta...
3. Procesamiento: el tratamiento digital de imágenes implica procesos que normalmente se expresan en forma de algoritmos por lo que necesitaremos hardwares específicos o bien ordenadores workstations.
4. Comunicación: comunicaciones locales entre sistemas de procesamiento de imágenes y comunicaciones remotas entre dos puntos, habitualmente para la transmisión de los datos de las imágenes.

---

# **Parte 2:**

# **Tratamiento digital de imágenes**

---



## **2. TRATAMIENTO DIGITAL DE LA IMAGEN**

### **2.1. FUNDAMENTOS DE LA IMAGEN DIGITAL**

En este punto analizaremos mecanismos de la percepción humana, con conceptos básicos de la formación de la imagen en el ojo y la capacidad de discriminar entre diferentes niveles de iluminación. Introduciremos también el concepto de muestreo uniforme de la imagen y cuantificación de niveles de gris, además de relaciones básicas entre píxeles y geometría de la imagen.

#### **2.1.1. PERCEPCIÓN VISUAL**

El objetivo final de muchas técnicas es que el observador interprete el contenido de cierta imagen, por lo que es de vital importancia una previa comprensión básica de los procesos de percepción visual.

Las técnicas de procesado de imágenes se realizan desde la percepción humana la cual resulta muy subjetiva, por lo que desarrollar un entendimiento básico puede resultar un proceso arduo.

El ojo humano está compuesto por dos clases de receptores: conos y bastones. En cada ojo existen entre 6 y 7 millones de conos situados principalmente en la parte central de la retina, nos permiten ver el color, y tenemos tres tipos: rojos, verdes y azules. La visión humana mediante los conos se denomina fotópica o visión de luz brillante. Por otro lado, el número de bastones es mucho mayor: entre 75 y 150 millones, éstos son sensibles a la luminancia y están situados a la periferia de la retina. Los bastones sirven para dar una visión general del campo de visión. Como tenemos más bastones que conos, nuestro ojo es mucho más sensible a las variaciones de luminancia que a las de crominancia, por lo que aprovechamos esto para comprimir las imágenes con mayor efectividad.

Respecto a la formación de imágenes en el ojo, el equivalente de la lente es el cristalino con la diferencia que éste es flexible. Para enfocar objetos lejanos, se aplana, y para enfocar objetos cercanos se ensancha. La percepción tiene lugar cuando los diferentes receptores son excitados, éstos transforman la energía radiante a impulsos eléctricos que son enviados al cerebro para su decodificación.

### 2.1.2. MUESTREO Y CUANTIFICACIÓN

Las imágenes reales son continuas por lo que para trabajar con ellas por ordenador deberemos digitalizarlas, es decir, discretizarlas. Este proceso de digitalización consta principalmente de dos fases: muestreo y cuantificación. El muestreo consiste en tomar puntos de la señal continua analógica a una frecuencia o tasa de muestreo constante, estos puntos en los que se integra el área son los elementos más pequeños en que se divide una imagen: los píxeles. Una vez realizado este proceso será necesario codificar digitalmente el color integrado en cada píxel, esta codificación se denomina cuantificación.

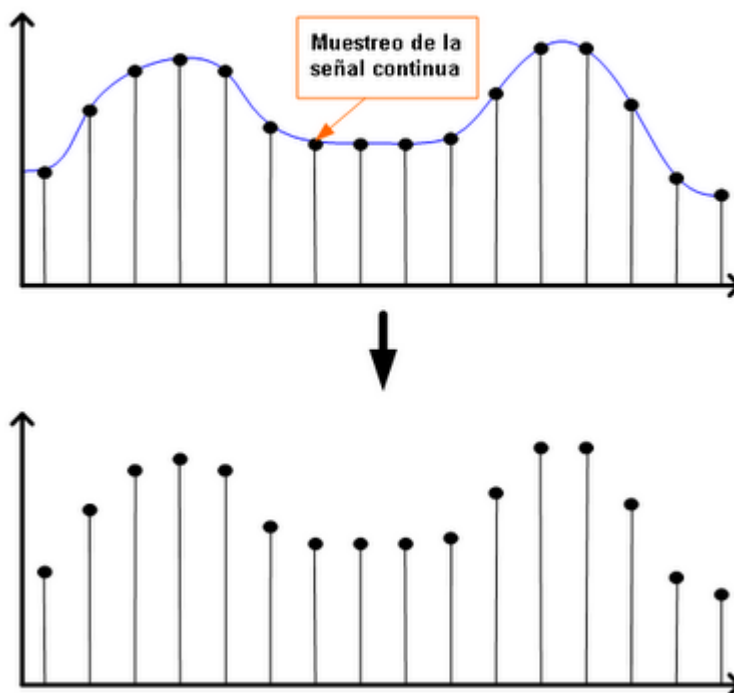


Figura 2.1.2.1. Muestreo de una señal.

Como cuantificación entendemos el proceso de convertir la sucesión de muestras de amplitud que hemos obtenido con el muestreo en una sucesión de valores discretos preestablecidos según el código utilizado. El fin de la cuantificación es cambiar la resolución de color de una imagen con una distorsión mínima. A la hora de codificar digitalmente una imagen es fundamental el número de bits utilizados para ello, ya que el

número de niveles que podemos obtener (N) depende de esta manera del número de bits empleados (m):  $N = 2^m$

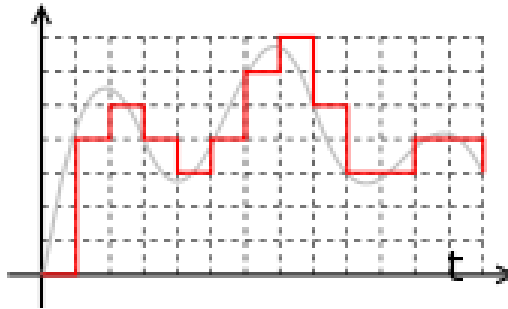


Figura 2.1.2.2. Señal cuantificada con 8 niveles, es decir, con 3 bits.

Los colores de la imagen se obtienen a partir de los primarios, normalmente: rojo, verde y azul (RGB), así pues, la codificación de color de un píxel se realizará dando un determinado peso a cada una de las componentes. Por lo tanto debemos tener en cuenta el número de bits que asignamos para codificar cada color, ya que tendremos que guardar nuestra imagen en un medio de almacenamiento, y deberemos tener cuidado con el tamaño final de ésta.

Si reducimos mucho el número de bits empleados obtendremos una pérdida de calidad considerable, y pueden llegar a aparecer falsos contornos, es decir, bordes inexistentes en la imagen original, también llamados “artefactos”.

### 2.1.3. RELACIONES BÁSICAS ENTRE PÍXELES

En este apartado nombraremos algunas de las relaciones entre píxeles más importantes, y para ello, nombraremos cada píxel con las letras minúsculas “p” y “q”.

Para facilitar la detección de bordes y áreas de objetos definimos la conectividad entre píxeles de una imagen.

Si consideramos V como el conjunto de valores de nivel de gris empleados para definir la conectividad podemos realizar la siguiente clasificación:

- 4-Conectividad:  
Un píxel tiene cuatro vecinos, dos horizontales y dos verticales, si éste se encuentra en el borde puede que algunos de sus vecinos caigan fuera de la imagen.

Cada uno de los 4 píxeles conectados al píxel en cuestión, se dice que están 4-conectados y que son 4-vecinos de éste.

Dos píxeles  $p$  y  $q$  con valores dentro de  $V$  están 4-conectados si  $q$  pertenece a los 4 vecinos de  $p$ .

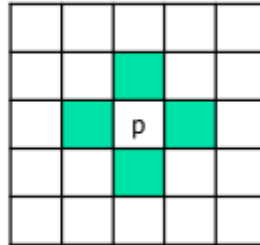


Figura 1.1.2.1. Ejemplo de 4-conectividad

En la figura superior podemos ver un ejemplo de 4-conectividad, dónde los píxeles coloreados en azul están 4-conectados y son 4-vecinos del píxel  $p$ .

- 8-conectividad:

Cuando hablamos de 8-conectividad englobamos la 4-conectividad, ya que ésta se compone de 4-vecinos además de los píxeles de sus diagonales.

Dos píxeles  $p$  y  $q$  con valores dentro de  $V$  están 8-conectados si  $q$  pertenece a los 8 vecinos de  $p$ .

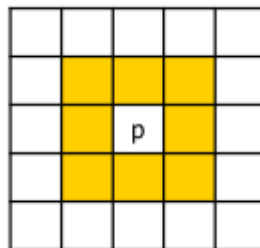


Figura 1.1.2.2. Ejemplo de 8-conectividad.

En la figura de arriba observamos que en este caso los píxeles amarillos serán los 8-vecinos del píxel  $p$ , y entre ellos formarán la 8-conectividad.

- m-conectividad:

Definimos  $m$ -conectividad como una conectividad mixta entre las dos nombradas anteriormente.

Dos píxeles  $p$  y  $q$  con valores dentro de  $V$  están  $m$ -conectados si:

- $q$  pertenece a los 4 vecinos de  $p$ , o bien
- $q$  pertenece a los 4 vecinos en diagonal del  $p$  y además el conjunto de la intersección de los píxeles 4 vecinos de  $p$  y los 4 vecinos de  $q$  es

vacío. (Este es el conjunto de píxeles que son 4-vecinos de  $p$  y  $q$  cuyos valores están en  $V$ .)

Para entender mejor las conectividades veremos la siguiente figura:

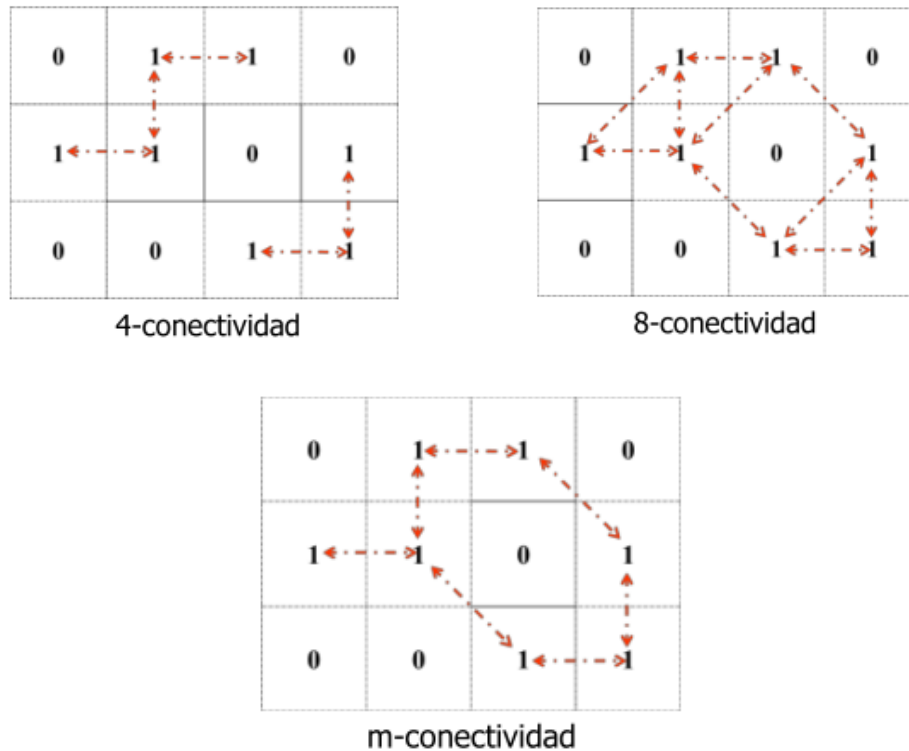


Figura 1.1.2.3. Ejemplo de conectividades.

La  $m$ -conectividad aparece para eliminar los múltiples caminos que aparecen al emplear la 8-conectividad.

#### 2.1.4. AJUSTES GEOMÉTRICOS

Son ajustes necesarios para poder comparar los píxeles de una imagen cuando existen fenómenos como ligeros movimientos que pueden alterar la intensidad de un píxel y por tanto detectar cambios que no nos interesan.

Para ello procederemos a la georreferenciación de imágenes, es decir, podremos la imagen en conformidad geométrica con la realidad en un sistema de proyección dado. Podemos conocer la posición de cualquier píxel si conocemos el número de filas y columnas y la dimensión y forma de los píxeles.

Es necesario realizar un re-muestreo de la imagen para volver a una representación sin discontinuidades después del proceso de georreferenciación, ya que éste deforma los píxeles de origen.

Este proceso se puede realizar principalmente en tres etapas:

- Decisión de la geometría de los píxeles de la imagen, tamaño, forma y coordenadas en el plano de proyección.
- Definir los parámetros necesarios para el cálculo de los píxeles de salida, además de asignarles un valor derivado al de los píxeles de la imagen original.
- Por último, conformar la relación entre píxeles georreferenciada mediante técnicas de mosaiqueado e indexación.

Dependiendo de la toma de las imágenes pueden haber diferentes causas de deformación de la imágenes, tales como: condiciones atmosféricas, posición durante la toma, calidad y deformación de la hoja en el caso que ésta sea escaneada, etc.

Existen sistemas de referencia a la hora de corregir geoméricamente una imagen. Son modelos que determinan la deformación entre las coordenadas de la imagen original y las coordenadas en el sistema de referencia utilizado para la relación a crear. Existen tres casos:

- Se conocen con gran precisión los modelos de deformación, por lo que bastará con realizar el modelo de la rectificación en base a las deformaciones conocidas.
- No se conocen los modelos de deformación. En este caso realizaremos modelos aleatoriamente de las deformaciones con una función dada y calcularemos los coeficientes del modelo en base a unos puntos de coordenadas conocidas.
- Se conocen de manera insuficiente los modelos de deformación. Se ajustan los modelos en base a los puntos de control (de coordenadas conocidas), el número de puntos depende de cada modelo.

En el proceso de re-muestreo nombrado anteriormente, calcularemos el valor para cada píxel de salida en función de los valores que tenían en la imagen original.

## **2.2 MEJORA DE LA IMAGEN**

Los procesos de mejora de la imagen consisten en un conjunto de técnicas que tratan de mejorar el aspecto visual de una imagen o convertir la imagen a una forma adecuada para el análisis de un ser humano o una máquina. Un ejemplo de mejora de una imagen podría ser resaltar los bordes de los objetos en una imagen por el filtrado de alta frecuencia.

No hay ninguna teoría general unificadora de mejora de la imagen en la actualidad porque no hay una norma general de calidad de imagen que pueda servir como un criterio de diseño para un procesador de mejora de la imagen.

Un método interesante para una aplicación puede ser inútil para otra.

### **2.2.1. PROCESADO DE PUNTO**

Este procesado es un método que trabaja directamente en el dominio de los píxeles. Dentro de él podemos realizar diferentes efectos para la modificación de la imagen según nuestra conveniencia.

#### **2.2.1.1. TRANSFORMACIÓN DE INTENSIDAD**

En este punto nombraremos algunas transformaciones básicas de las imágenes dentro de las transformaciones de intensidad.

En primer lugar nombraremos el negativo de una imagen, este método trata de hacer la inversa de cada pixel de una imagen, es decir, si por ejemplo un pixel es negro lo cambiaremos a blanco y así sucesivamente con cada uno de los píxeles correspondientes a la imagen.



Figura 2.2.1.1.1. Ejemplo transformación negativo.

Otra transformación muy común es el cambio de brillo de una imagen, es decir, el cambio de luminosidad. El máximo valor del brillo es una imagen blanca, y el mínimo es una imagen negra. A continuación podemos observar un una imagen modificada por el brillo.



Figura 2.2.1.1.2. Ejemplo de modificación de brillo de una imagen.

Cuando hablamos de contraste de una imagen nos referimos a la diferencia entre los puntos más oscuros y más claros de la imagen.



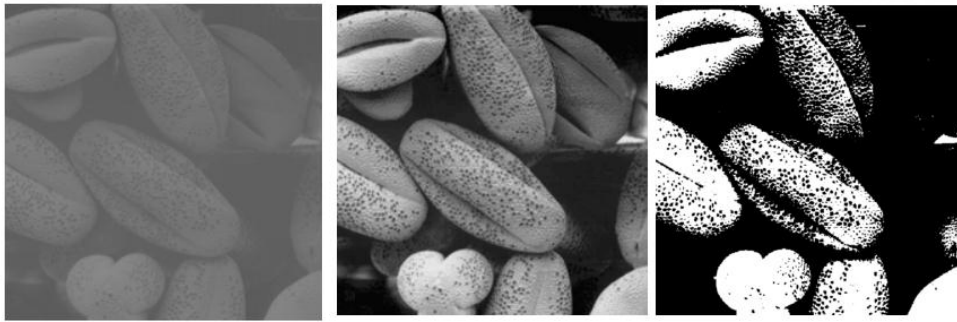


Figura 2.2.1.1.3. Ejemplo de modificación de contraste de menos a más.

Por último, la compresión del rango dinámico se utiliza para poder diferenciar niveles de gris muy alejados y el rebanado del nivel de gris, para resaltar un rango concreto de niveles de gris de una imagen.

### 2.2.1.2. PROCESADO DEL HISTOGRAMA

El histograma es una representación gráfica en forma de barras, que muestra el número de píxeles de una imagen que tiene cada nivel de gris (en el caso de una imagen en blanco y negro). Da una idea de la distribución de niveles de gris en una imagen y permite ver de forma clara si la imagen es demasiado brillante, oscura o poco contrastada. El perfil del histograma de una imagen proporciona una información muy útil sobre la posibilidad de mejora de la imagen.

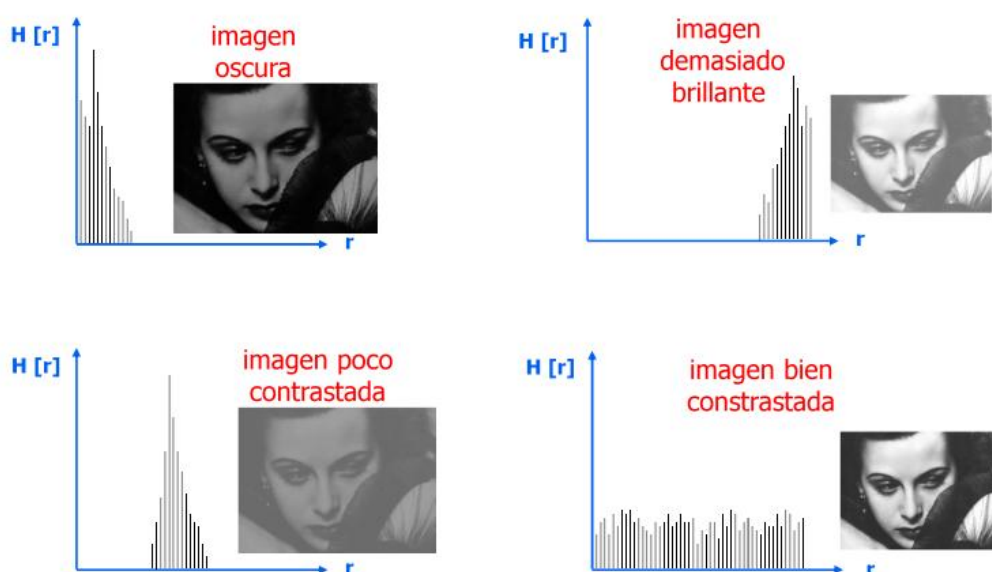


Figura 2.2.1.2.1. Histogramas más comunes.

Así pues existen diferentes métodos y funciones para conseguir un histograma ecualizado que nos permita obtener la imagen deseada.

### 2.2.1.3. SUBSTRACCIÓN DE IMÁGENES

Se trata de realizar la diferencia de niveles de gris entre dos imágenes para ver objetos en movimiento. Este procedimiento es muy utilizado en técnicas medicinales como la radiografía.

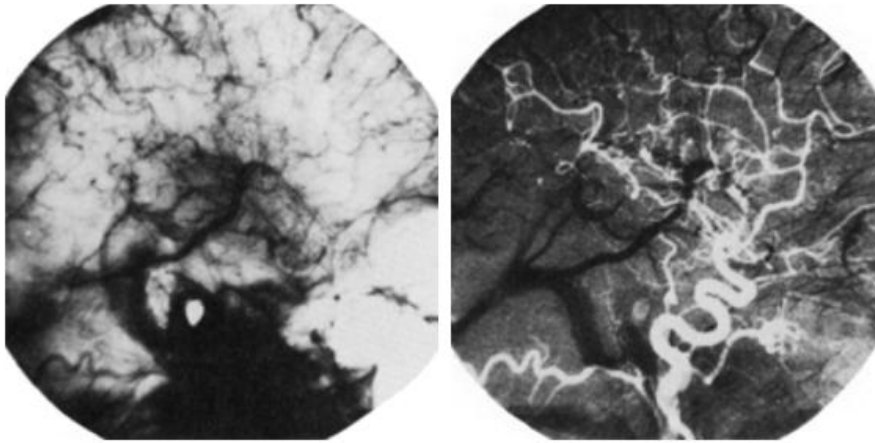


Figura 2.2.1.3.1. Ejemplo de substracción de imágenes.

En la Figura 2.3.1.3.1 encontramos un ejemplo de mejora de imágenes mediante la técnica de substracción. Las imágenes son iguales, con la diferencia que la imagen de la derecha ha sido tomada un instante después tras la inyección de un líquido en el torrente sanguíneo.

### 2.2.1.4. MEDIA DE IMÁGENES

Es una técnica utilizada para la eliminación de ruido, realizando un promediado de varias imágenes del mismo objeto tomadas en diferentes instantes de tiempo.

Una importante aplicación de un promedio de imagen está en el campo de la astronomía, dónde las imágenes que se toman normalmente tienen niveles de luz muy

bajos y con ruido causados por los sensores, por lo que la toma de una sola imagen es prácticamente inútil para el análisis.

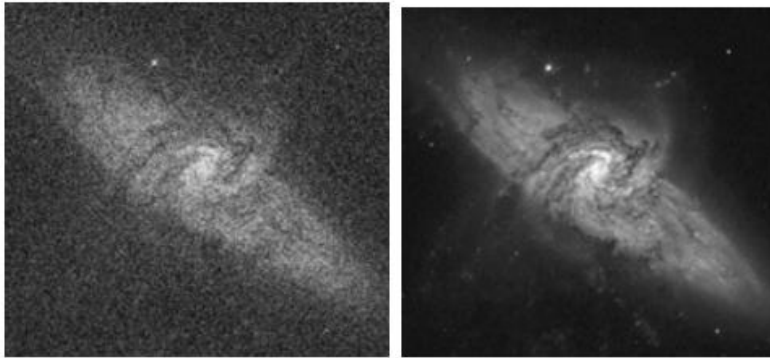


Figura 2.3.1.4.1 Ejemplo del promediado de imágenes en astronomía.

## 2.2.2. FILTRADO ESPACIAL

### 2.2.2.1. FILTROS DE SUAVIZADO

Son filtros cuya característica común es que disminuyen las diferencias de los valores de los píxeles respecto a sus vecinos. La aplicación de este tipo de filtros produce la degradación de la imagen, un efecto que se asocia al desenfoque de la imagen debido a la difuminación de las líneas de contorno de los objetos.

Atenúan las altas frecuencias dejando pasar las bajas, aparte de difuminar la imagen también se utilizan para reducir el ruido de una imagen. A pesar de que en un principio no veamos las ventajas de hacer borrosa una imagen, esto puede ser muy útil en el procesado para eliminar pequeños detalles antes de la extracción de un objeto grande y el relleno de pequeños espacios entre líneas o curvas.

Los tres principales filtros de suavizado son: el de media, el filtro Gaussiano y el filtro de la mediana.

El filtro de media consiste en asignar al píxel central de la imagen filtrada el valor correspondiente a la media aritmética del valor de gris del píxel central y sus 8 vecinos para un filtro de 3x3, los 24 vecinos cuando el filtro es 5x5 y así sucesivamente. Aumentado el valor del píxel central del filtro disminuiríamos el efecto de suavizado.

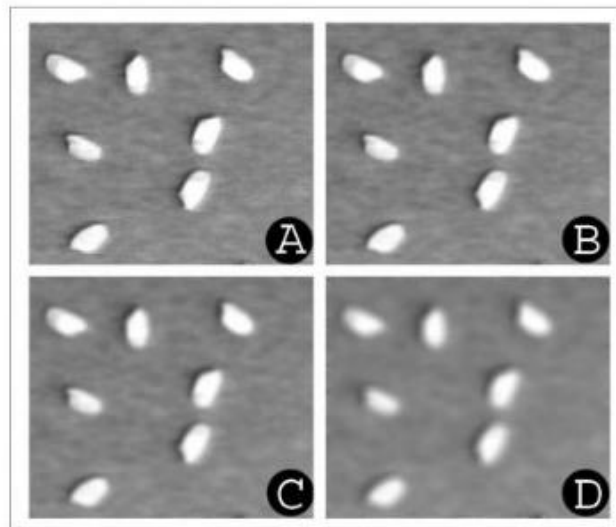


Figura 2.3.1.1.1. Aplicación filtro de media.

Los filtros Gaussianos producen un efecto parecido al de los filtro de media, pero aquí si se le da más peso al píxel central que a sus vecinos, a la vez que se le da más peso a los píxeles más próximos y menos a los píxeles más alejados. El peso del píxel depende de la distancia al píxel central.

$$\mathbf{G}_{3 \times 3} = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}; \quad \mathbf{G}_{7 \times 7} = \begin{bmatrix} 1 & 3 & 7 & 9 & 7 & 3 & 1 \\ 3 & 12 & 26 & 33 & 26 & 12 & 3 \\ 7 & 26 & 55 & 70 & 55 & 26 & 7 \\ 9 & 33 & 70 & 90 & 70 & 33 & 9 \\ 7 & 26 & 55 & 70 & 55 & 26 & 7 \\ 3 & 12 & 26 & 33 & 26 & 12 & 3 \\ 1 & 3 & 7 & 9 & 7 & 3 & 1 \end{bmatrix}$$

Figura 2.3.1.1.2. Ejemplos de máscaras Gaussianas.

Por último en el filtro de mediana el nuevo valor del píxel central no se obtiene como promedio de los nueve vecinos, sino tras ordenar los valores y buscar el valor de la mediana en la serie, es decir el valor que queda en el centro. Es muy utilizado para la eliminación de manchas de ruido de fondo.

### 2.2.2.2. FILTROS DE REALZADO

Estos filtros son utilizados cuando el contraste entre los objetos y el fondo sea muy bajo y sea difícil la segmentación directa de los objetos. Destacan los detalles finos e intensifican los detalles difuminados por error o por el método de adquisición.

Las máscaras de estos filtros se caracterizan por tener coeficientes positivos en el centro y coeficientes negativos en la periferia, por lo que la aplicación de éstos puede dar lugar a valores negativos o mayores de 255 y será necesario un re-escalado del histograma de niveles de gris para prevenir este fenómeno.

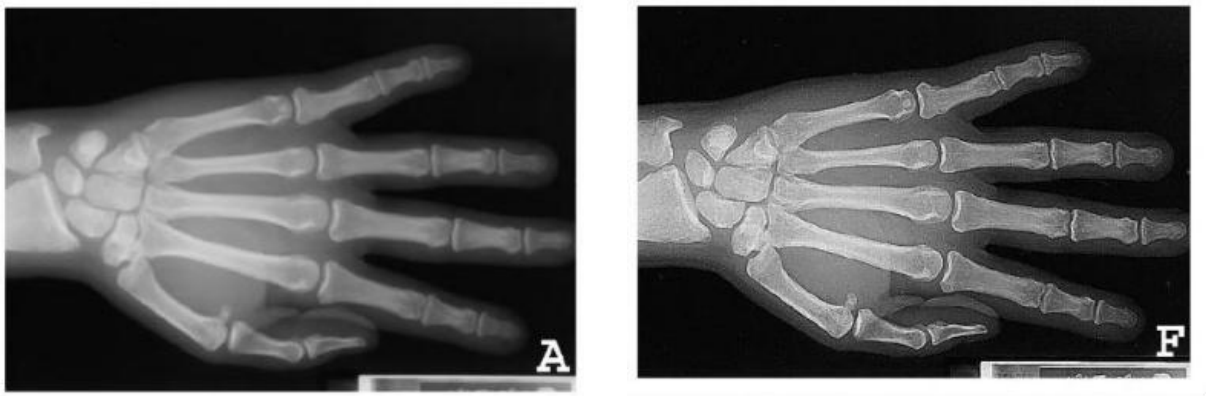


Figura 2.2.2.2.1. Ejemplo de filtro de realce.

### 2.2.3. DOMINIO DE LA FRECUENCIA

Se diseña el filtro directamente en el dominio de la frecuencia y realizaremos el filtrado mediante el producto de las transformadas de Fourier de la imagen y el filtro. Este producto se realiza multiplicando frecuencia a frecuencia de cada elemento teniendo en cuenta la propiedad de bidimensionalidad.

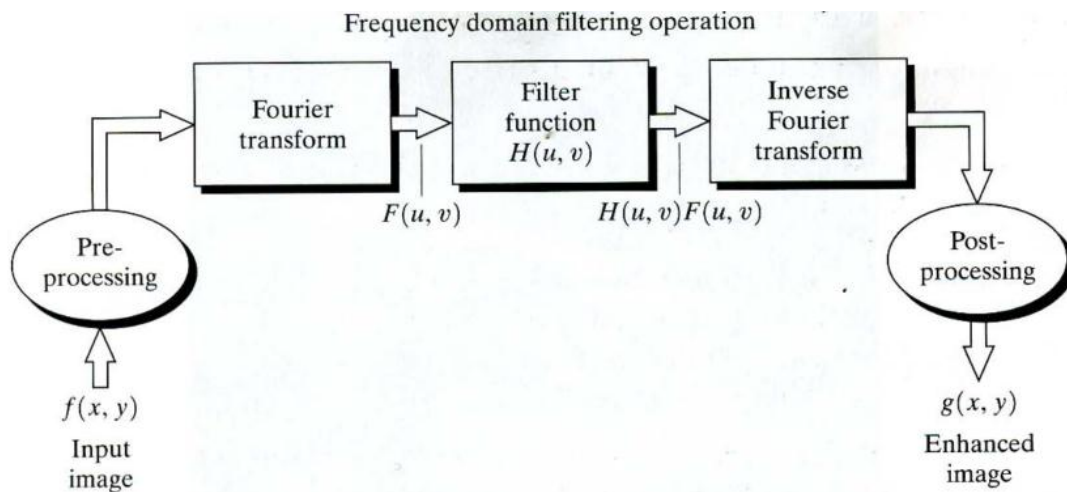


Figura 2.2.3.1. Pasos básicos para el filtrado en el dominio de la frecuencia.

En nuestro caso hablaremos de la transformada de Fourier discreta, que no es más que un conjunto de muestras de la transformada de Fourier de la imagen digital. Como las imágenes son bidimensionales realizaremos esta transformada tanto de las filas como de las columnas, es decir, tendremos dos sumatorios en vez de uno.

Otro de los bloques nombrados en la figura 2.3.3.1 es la transformada de Fourier inversa, que nos permite recuperar la señal a partir de su transformada.

A la hora de visualizar la transformada discreta de Fourier debemos tener en cuenta que ésta tiene un rango de valores muy grande por lo que solo serán visibles los valores muy brillantes. Para la correcta visualización lo que haremos será realizar una compresión logarítmica de los valores.

Así pues, podemos ser capaces de realizar los filtros nombrados anteriormente, de realzado o suavizado, en el dominio de la frecuencia.

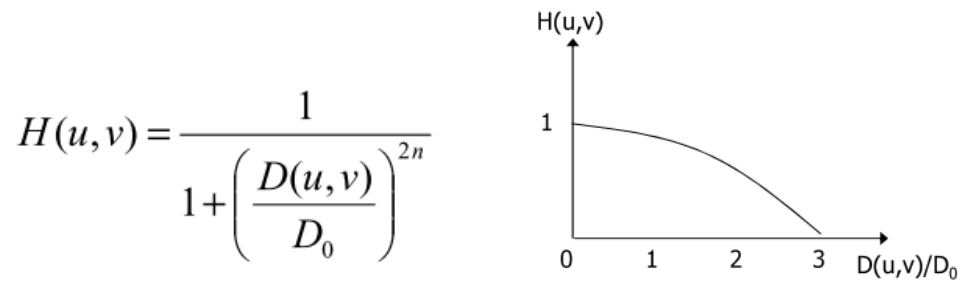


Figura 2.2.3.2. Ejemplo de filtro paso bajo o de suavizado en el dominio de la frecuencia.

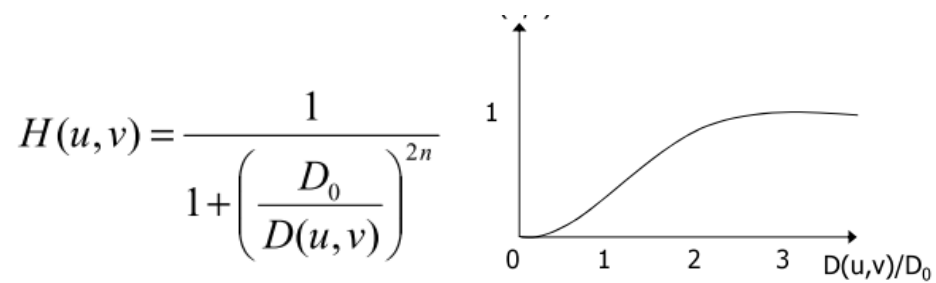


Figura 2.2.3.3. Ejemplo de filtro paso alto o de realce en el dominio de la frecuencia.

## 2.3. RESTAURACIÓN DE IMÁGENES

La mejora de la imagen es en gran medida un proceso subjetivo, mientras que la restauración de la imagen es en mayor parte un proceso objetivo. La restauración intenta mediante el uso de un conocimiento a priori del fenómeno de la degradación, reconstruir o recuperar una imagen que ha sido degradada. Por lo tanto las técnicas de restauración están orientados hacia el modelado de la degradación y aplicando el proceso inverso con el fin de recuperar la imagen original.

Existen métodos de restauración que trabajan en el dominio de la frecuencia y otros que trabajan en dominio espacial.

En este punto del procesado digital de imágenes trataremos de buscar los filtros que han generado la degradación de la imagen, para multiplicar su inversa con la imagen degradada y así recuperar la imagen restaurada.

En una degradación por movimiento la transformada de Fourier tendrá ceros que estarán más juntos conforme más grande sea el desplazamiento. Aparecerán unas líneas verticales negras cuando el movimiento que se ha producido ha sido horizontal.

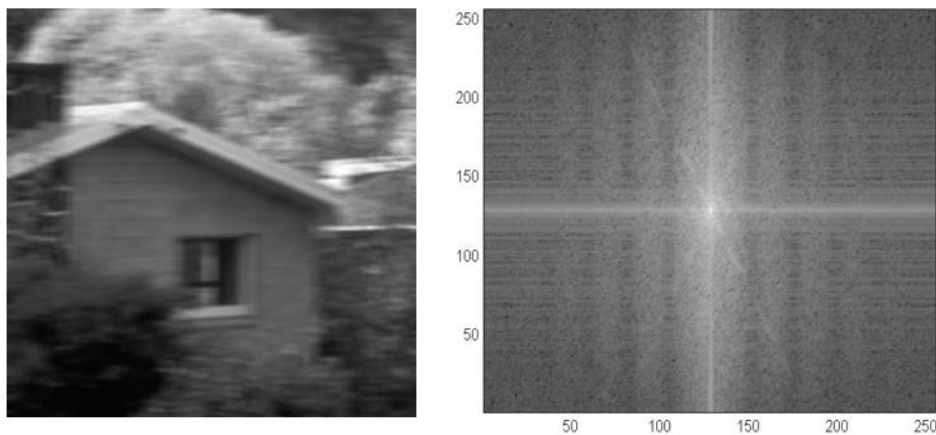


Figura 2.3.1. Restauración de imágenes

Un proceso común a la hora de realizar una restauración de una imagen podría ser el siguiente:



- Encontrar mediante técnicas de tanteo o cepstrales, la mejor aproximación posible a la degradación que ha ocurrido.
- Añadiremos ceros hasta que el tamaño del filtro sea el mismo que el tamaño de la imagen degradada.
- Calcular la FFT del supuesto filtro degradador y de la imagen degradada.
- Deconvolucionar multiplicando la FFT de la imagen por la inversa del filtro degradador.
- Realizar la FFT inversa.

Cuando se trata de desenfocados de la imagen la máscara que modela el desenfocado es parecido a un círculo, de manera que ha mayor tamaño del círculo mayor será el desenfocado. En cambio cuando la degradación se ha producido por interferencias el proceso cambia. En el dominio de la frecuencia una señal senoidal se ve como un pico muy alto de dicha transformada por lo que para eliminarla se puede utilizar un filtro de banda eliminada que deje toda la transformada igual excepto la frecuencia de la interferencia. Los pasos que seguiremos serán los siguientes:

- Realizar la FFT de la imagen a restaurar.
- Localizar los puntos brillantes de la interferencia en la transformada.
- Eliminarlos poniéndolos a cero.
- Realizar la transformada inversa.

Imagen con interferencias



Imagen restaurada



Figura 2.3.2. Imagen restaurada

Otro de los motivos por el que podríamos precisar de la restauración de imágenes podría ser el ruido. Hay diferentes tipos de ruido, pueden afectar a todos los píxeles de la imagen o solo a parte de ellos, dependiendo del tipo que estemos tratando emplearemos un tipo de filtro u otro.

Por último, podría darse el caso que la imagen estuviera mal iluminada y tengamos que corregir este fallo para poder procesar la imagen correctamente, para ello deberemos saber que la función de iluminación es de baja frecuencia y está sumada a la función de reflectividad. Para disminuir el efecto de la iluminación una opción es atenuar las bajas frecuencias, siempre y cuando respetemos la continua.

## 2.4. SEGMENTACIÓN

Mediante el proceso de segmentación trataremos de dividir la imagen en las partes u objetos que la forman. El nivel de subdivisión que realicemos dependerá de la cada aplicación en particular, es decir, la segmentación terminará cuando se hayan detectado todos los objetos de interés para dicha aplicación. Realizar una segmentación automática es uno de los procesos más complicados dentro del procesado de imágenes, ya que durante este proceso corre la responsabilidad de obtener una buena solución o no.

Existe una gran variedad de técnicas de segmentación, basadas en diferentes técnicas, que se adaptan al tipo de imagen y al objeto que se persigue. Principalmente podemos destacar:

- Mediante umbralización
- Mediante detección de regiones
- Mediante detección de fronteras

### 2.4.1. SEGMENTACIÓN POR UMBRALIZACIÓN

Supongamos una imagen donde los niveles de gris están agrupados en dos modos predominantes. Una forma de separar los objetos del fondo consiste en seleccionar un umbral que separe estos modos donde cada modo pertenecerá a una parte de la imagen. En la figura 2.4.1.1 vemos un ejemplo donde  $T$  es el umbral fijado y la parte de la izquierda, por ejemplo, podría ser el fondo y la parte de la derecha el objeto. Así pues se puede dar el caso de determinar no solo un umbral, sino varios.

Por lo que hemos visto una parte fundamental para este tipo de segmentación es el histograma. Otra parte fundamental es la elección del umbral, ésta puede hacerse de modo manual, tras la visualización de la imagen y su histograma, o de forma automática. En imágenes donde el histograma presenta agrupaciones claras, la segmentación por umbralización podrá realizarse de forma clara y fácil; en cambio, cuando las agrupaciones no sean claras este método no podrá utilizarse.

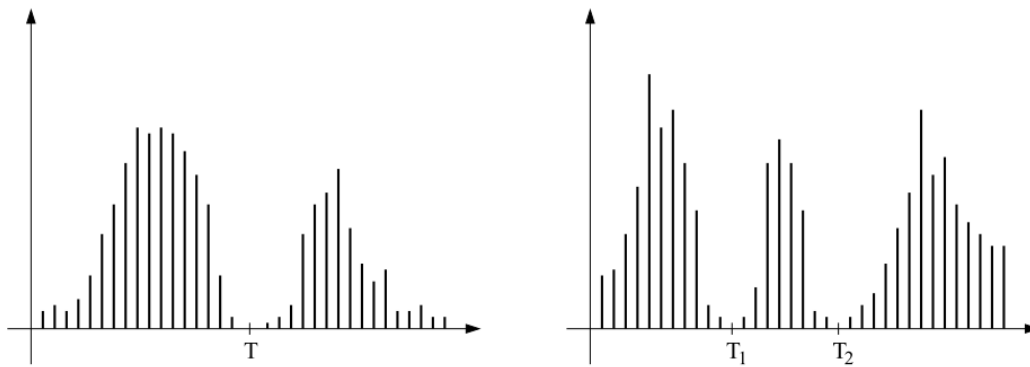


Figura 2.4.1.1. Ejemplo de histograma donde se utiliza un umbral o varios.

Dentro de la segmentación por umbralización, cabe destacar los posibles efectos por iluminación de la imagen, que hacen que ésta no sea tan sencilla, o la umbralización de texturas. En la umbralización por texturas, debemos tener en cuenta que dependiendo de la textura que nos encontremos deberemos realizar un filtrado de media o de varianza previo a la umbralización. Si la textura posee un patrón de repetición, la mejor opción será recurrir al dominio de la frecuencia.

## 2.4.2. SEGMENTACIÓN POR DETECCIÓN DE REGIONES

En este tipo de segmentaciones al contrario de la nombrada anteriormente, tendremos en cuenta las propiedades espaciales de la imagen. Se trata de métodos iterativos en los que parten de una situación inicial y se va refinando mediante crecimiento de regiones o por el método de partir y unir.

Como región entendemos un conjunto de píxeles conectados entre mediante alguna de las conexiones nombradas anteriormente.

Como hemos dicho, dos de las principales técnicas que se emplean son las siguientes;

- Crecimiento de regiones: partimos de un primer punto inicial, elegido manual o automáticamente, e iremos agregándole los píxeles vecinos que cumplan una determinada condición, como puede ser que tengan un nivel de gris similar. Así pues, la región irá creciendo progresivamente hasta obtener el número de puntos

que conformen el objeto deseado. Partiremos de tanto puntos como objetos queramos segmentar.

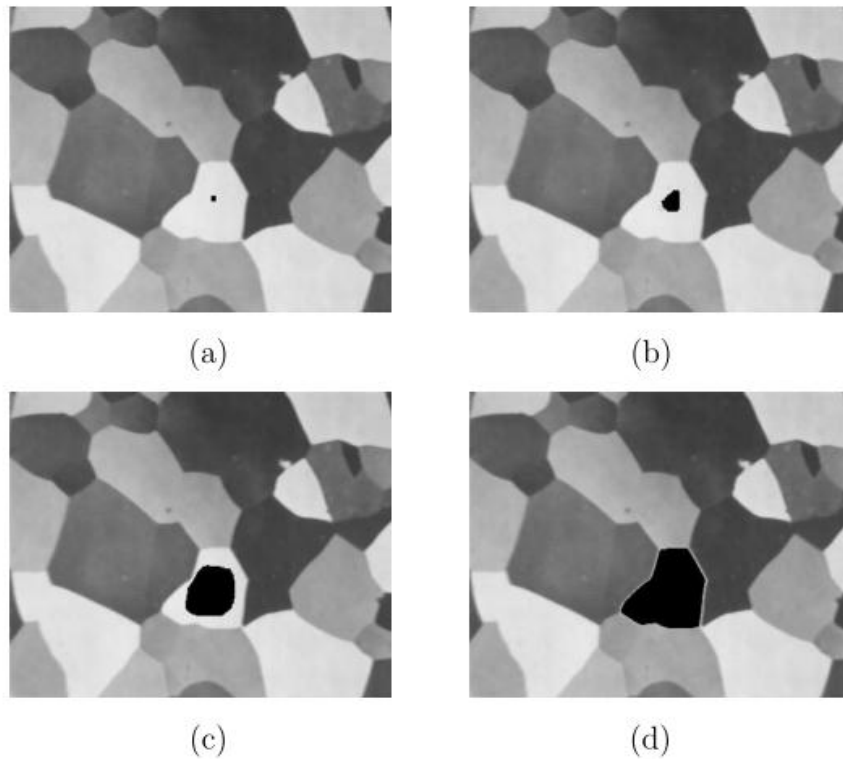


Figura 2.4.2.1. Ejemplo de utilización del método crecimiento de regiones.

- Partir y unir: se trata de subdividir la imagen en un conjunto arbitrario de regiones disjuntas y posteriormente fusionar y/o dividir estas regiones con el objetivo que finalmente se cumpla la geometría del objeto deseado, es decir, seguir dividiendo en subregiones hasta que cada región cumpla el criterio de homogeneidad marcado. Al mismo tiempo se comprueba, tras cada subdivisión, si cualquiera de las regiones puede ser unida a una de sus regiones adyacentes para dar lugar a una región mayor que cumpla el criterio de homogeneidad. En general, la división en subregiones es siempre una división en cuadrantes, de forma que cualquier región no homogénea se descompondrá en cuadrantes iguales.

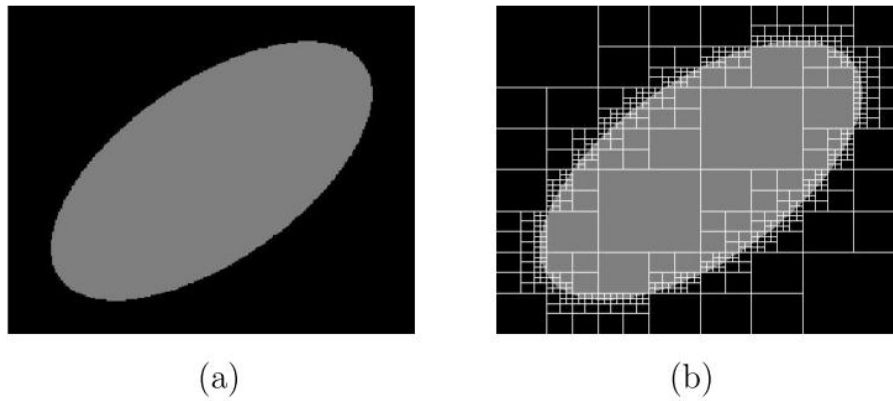


Figura 2.5.1.2. Ejemplo del método partir y unir.

### 2.4.3. SEGMENTACIÓN POR DETECCIÓN DE FRONTERAS

Esta tipo de segmentación también se suele llamar detección de discontinuidades ya que se basan en buscar discontinuidades en la imagen, es decir, píxeles correspondientes a fronteras entre regiones, para, una vez localizados éstos, identificar las diferentes regiones separadas por dichas fronteras.

Para ello emplearemos diferentes técnicas como son, la detección de puntos, de bordes, de líneas y la unión de bordes.

- Detección de puntos: aplicaremos una máscara a cada punto de la imagen, produciendo un valor 'R', y con este valor determinaremos un umbral de forma que si el módulo de 'R' es mayor a éste, habremos detectado un punto. El tamaño de los puntos detectados puede variar con el tamaño de la máscara empleada, pueden ser puntos de un píxel, de 2x2 píxeles, etc. Un ejemplo de máscara de detección de punto de tamaño un píxel es:

<b>-1</b>	<b>-1</b>	<b>-1</b>
<b>-1</b>	<b>8</b>	<b>-1</b>
<b>-1</b>	<b>-1</b>	<b>-1</b>

Figura 2.4.3.1. Ejemplo máscara.

- Detección de líneas: aplicaremos una máscara a cada punto de la imagen, produciendo un valor 'R' como en el caso anterior. La diferencia es que cuando el módulo de 'R' supere un cierto umbral estaremos ante una línea detectada. Un ejemplo de máscara utilizada para la detección de líneas horizontales podría ser la siguiente:

<b>-1</b>	<b>-1</b>	<b>-1</b>
<b>2</b>	<b>2</b>	<b>2</b>
<b>-1</b>	<b>-1</b>	<b>-1</b>

Figura 2.4.3.2. Ejemplo máscara.

Al igual que en la detección de puntos también podemos variar el tamaño de las líneas que vamos a detectar.

- Detección de bordes: para la detección de bordes utilizaremos gradientes, es decir vectores de dos dimensiones que contienen los gradientes en cada dirección. Los gradientes son las derivadas primeras de la función imagen.

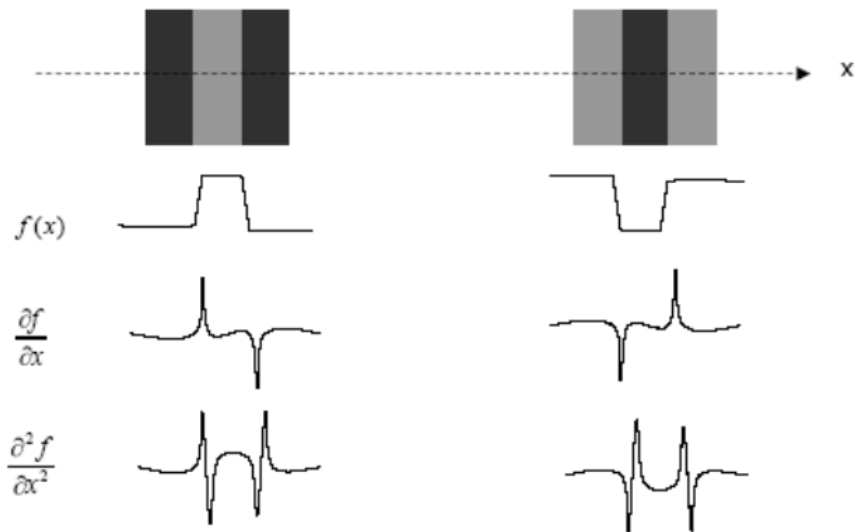


Figura 2.4.3.3. Gradiente de una imagen.

- Unión de bordes: en la umbralización del gradiente puede darse el caso de que algunos bordes queden recortados, por lo que será necesaria la unión de éstos. Para la unión se pueden emplear diferentes técnicas tales como la transformada de Hough o el procesamiento local.



## 2.5. REPRESENTACIÓN Y DESCRIPCIÓN

Estamos ante el paso siguiente a la segmentación. Después de haber segmentado la imagen y haber obtenido los objetos buscados, procederemos a identificarlos, es decir, describirlos a través de la medida de los parámetros del mismo. Mediremos las características individuales de cada objeto, y compararemos éstas con algunas medidas conocidas o patrones para poder clasificar un objeto en algunas de las categorías establecidas.

Podemos representar el objeto básicamente de dos formas: mediante sus características externas (contorno del objeto), o sus características internas. Dependiendo del método escogido para la representación utilizaremos distintos descriptores. Como descriptor entendemos una serie de características del objeto que nos sirven para determinar que objeto estamos tratando.

Según la clasificación que hemos hecho anteriormente de los tipos de representación, elegiremos una representación externa cuando la importancia del objeto está en su forma. Así pues, elegiremos una representación interna cuando la importancia recaiga en propiedades regionales como el color, brillo o textura.

Como ya hemos nombrado, existen diversos descriptores, de los cuales nombraremos algunos de ellos a continuación.

### 2.5.1. DESCRIPTORES DE FRONTERA

- Descriptores simples: perímetro y diámetro. El perímetro representa la longitud del borde del objeto y podemos se calcula de dos formas distintas según el tipo de conectividad. Asignaremos valor 1 si entre dos píxeles consecutivos existe 4-conectividad y  $\sqrt{2}$  para 8-conectividad. Por otro lado el diámetro de un objeto es la mayor distancia entre dos puntos del borde de éste, y al segmento que une estos puntos se le denomina eje mayor.
- Código cadena: describe el borde de un objeto basándose en su conectividad. Se establecen unos números de dirección con los que se recorre el perímetro del objeto y se crea un código de números. Sin embargo, este código depende del punto inicial, por lo que para evitar esto se genera el “Número de forma”. Por ello el procedimiento que seguiremos será en primer lugar, realizar una resta circular de cada número del código de cadena con el anterior, y posteriormente

se rota el código diferencial obtenido hasta obtener el número más pequeño posible.

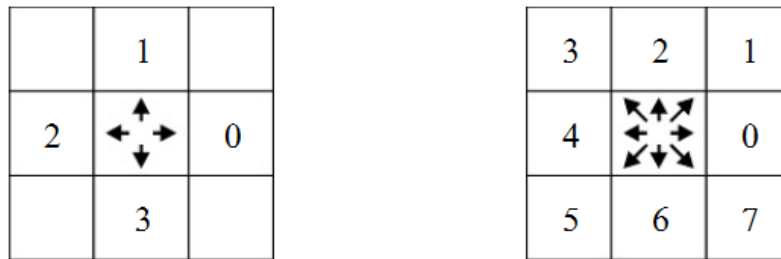


Figura 2.5.1.1. Numeración de dirección para 4-conectividad y 8-conectividad.

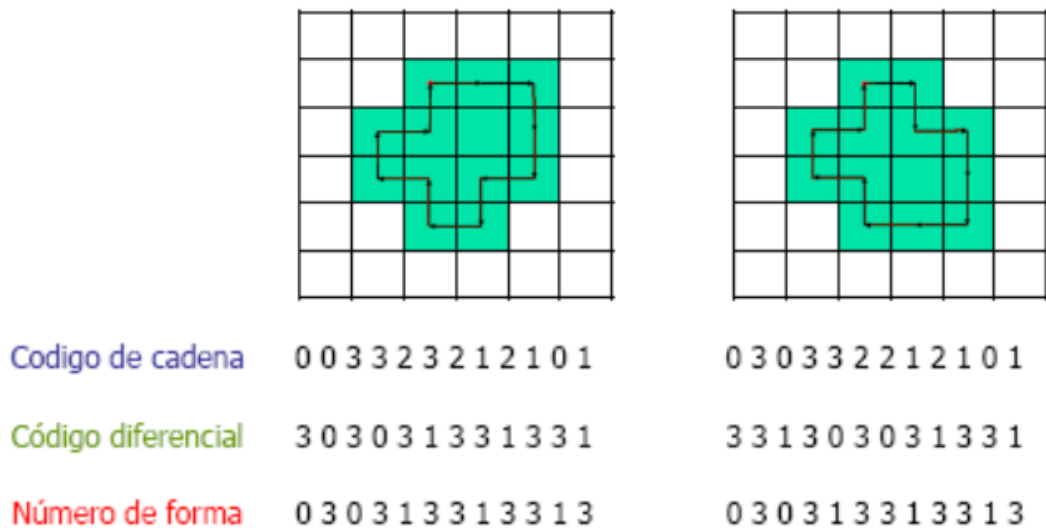


Figura 2.5.1.2. Ejemplo de cálculo del número de forma.

- Firmas: representan los bordes de un objeto en forma de función unidimensional. Se trata de método variante a la rotación y tamaño, por lo que existe una firma normalizada que permite que ésta sea invariante al tamaño.
- Aproximaciones poligonales: se caracteriza por aproximar el borde de los objetos mediante líneas. El procedimiento será sobre-imponer una rejilla al objeto y calcular una línea de perímetro mínimo que no atravesase ninguna de las

intersecciones de la rejilla. A continuación en la figura 2.6.1.3 podremos observar un ejemplo del empleo de esta técnica.

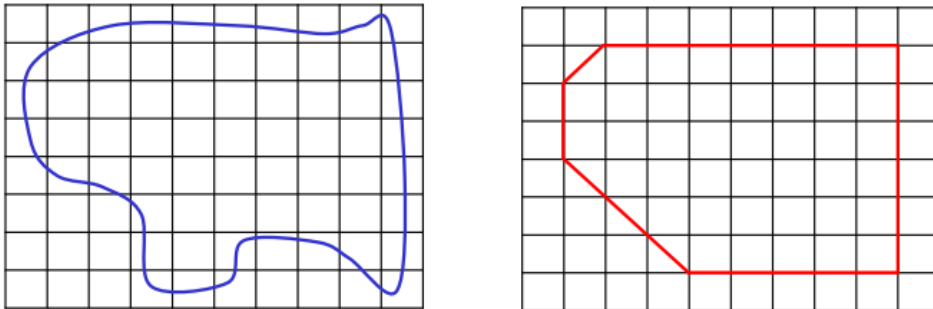


Figura 2.5.1.3. Ejemplo de aproximación poligonal.

- Segmentos de borde: segmentos que unen los puntos de contacto con el borde.
- Esqueletización: proporcionan un descriptor muy completo del objeto, pues se tiene información como la longitud de los segmentos y el número y posición de las bifurcaciones.

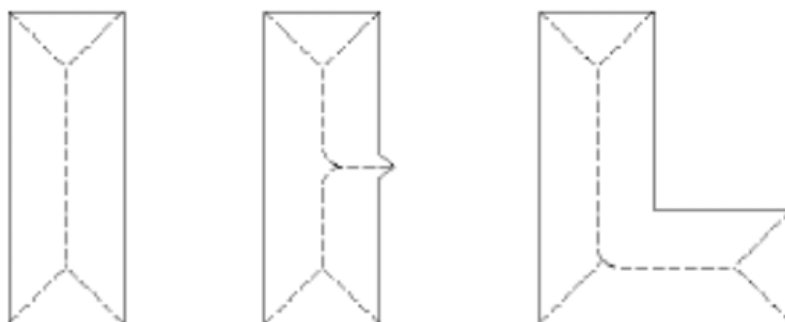


Figura 2.5.1.4. Ejemplo de esqueleto de figuras.

- Descriptores de Fourier: Trataremos los píxeles del borde de un objeto como si fueran puntos en el plano complejo. Sus valores se irán almacenando en un

vector y se realiza la transformada de Fourier del mismo. Los coeficientes de la transformada a cada frecuencia serán los que caractericen al objeto. Dependiendo del número de coeficientes que tomemos, la reconstrucción del objeto quedará más o menos definido. Estos coeficientes sirven también para detectar simetrías en los objetos. Los descriptores de Fourier nos suministran descriptores de forma invariantes, muy útiles para reconocer objetos.

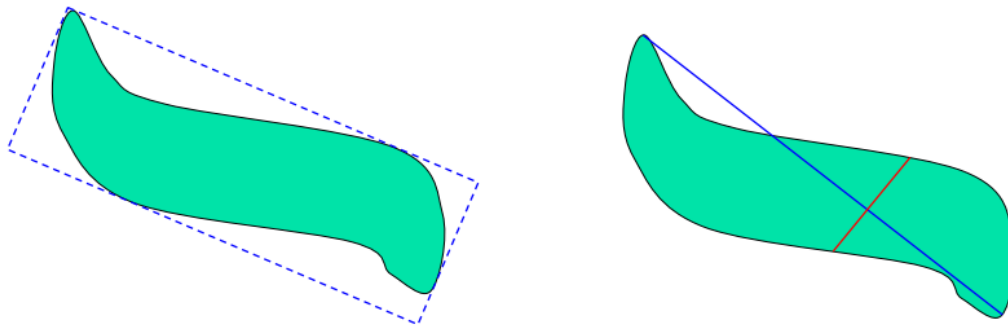
### 2.5.2. DESCRIPTORES DE REGIÓN

- Área: mide la extensión o el tamaño en píxeles normalmente de un objeto.
- Compacidad: descriptor que relaciona el área y el perímetro de un objeto. No tiene dimensiones y por tanto es insensible a los cambios de tamaño y orientación. Cuando el resultado es igual a la unidad se trata de círculo perfecto, y a partir de ahí puede variar su valor hasta el cero. A continuación podemos observar la fórmula de cálculo de la compacidad, donde “A” es el área del objeto, y “p” el perímetro.

$$C = \frac{4\pi A}{p^2}$$

Figura 2.5.2.1. Fórmula de la compacidad.

- Relación de aspecto: es la proporción entre el ancho y altura de un objeto. Pero hay que tener un cuenta que en un objeto se puede entender la altura y la anchura de diferentes modos. Por una parte la excentricidad relaciona el eje mayor con el eje menor, entendiendo éstos como el mayor y menor segmento que une los puntos del diámetro. Sin embargo, la elongación es el cociente entre los lados de la caja de área mínima que encierra a un objeto.



$$\text{Elong} = 9,32 / 2,48 = 3,75$$

$$\text{Excent} = 9,13 / 3,15 = 2,89$$

Figura 2.5.2.2. Ejemplo de cálculo de elongación y excentricidad.

- Descriptores topológicos: su característica principal es que no varían aunque deformemos el objeto, lo escalemos o cambiemos su perspectiva. El número de Euler se define como la resta del número de componentes conectados del objeto y el número de agujeros del objeto.
- Medidas de texturas: son utilizados cuando nos interesa más la distribución de los niveles de gris del objeto que su forma. Hay dos tipos:
  - Espectrales: se basan en las propiedades del espectro de la transformada de Fourier, ya que detectamos fácilmente mediante picos estrechos y de gran energía las periodicidades de una imagen.
  - Estáticos: se basan en el histograma de la imagen, como la media, varianza, entropía, etc.

### 2.5.3. DESCRIPTORES RELACIONALES

Como hemos visto hasta ahora, los descriptores mencionados se aplican a objetos individuales, en cambio los que veremos a continuación tienen un nivel superior de complejidad, ya que tratan de explotar la estructura existente entre partes de objetos más complejos.

A partir de una serie de funciones primitivas se construyen o modelan objetos complejos. Se va formando una cadena que define la estructura del objeto y que define en si misma un descriptor de dicho objeto.

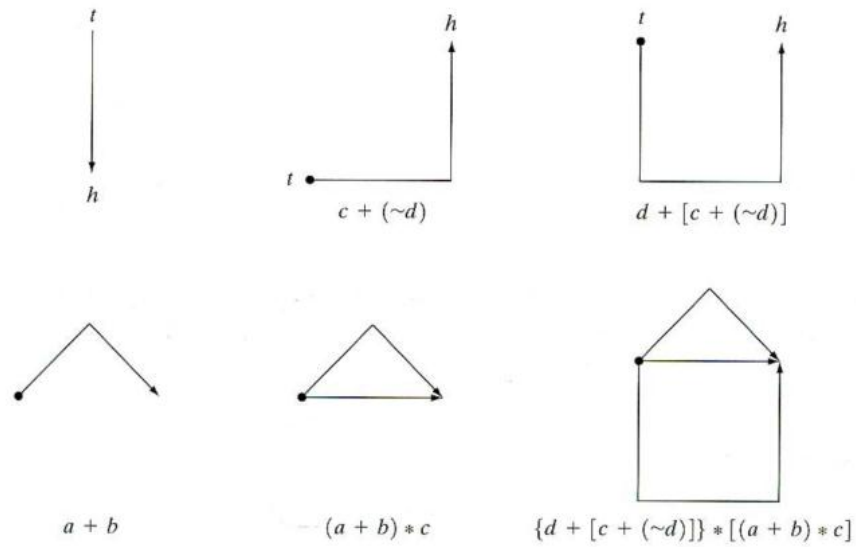


Figura 2.5.3.1. Ejemplo de descriptores relacionales.

## 2.6. RECONOCIMIENTO E INTERPRETACIÓN

Con esta parte concluiremos la breve introducción a las técnicas del procesado digital de imágenes. Esta es la última fase de un sistema de visión artificial, donde se trata de delegar en una máquina la habilidad humana de la visión.

Se trata de una tarea muy compleja, por lo que los sistemas que diseñan son muy específicos, para tareas muy concretas, como el reconocimiento óptico de caracteres o el control de calidad en una industria.

Empezaremos definiendo dos conceptos fundamentales en el reconocimiento de formas: formas y clases.

Una forma es una descripción cuantitativa o estructural de un objeto en una imagen, es decir, un conjunto de medidas. Un ejemplo podrían ser el perímetro, el área, la elongación, etc. Estas características se agrupan en vectores que caracterizarán y describirán cada uno de los objetos.

Por otro lado, las clases se definen como agrupaciones de formas que comparten unas características comunes y que se asocian a objetos. Las clases son denotadas como:  $\omega_1, \omega_2, \dots, \omega_N$ .

### 2.6.1. MÉTODOS DE DECISIÓN

Cuando llega la hora de decidir la asignación de una forma a una clase, el proceso puede llegar a complicarse. Puede no ser sencillo por diferentes motivos, la forma no es clara y evidente, no puede ser definida a priori o directamente no se conoce a que clase puede pertenecer. Por ello se utilizan diferentes estrategias que nombraremos a continuación.

Para clasificar las formas a las distintas clases se determinan unas zonas, pero cuando éstas no se pueden definir explícitamente se recurre a lo que se denomina un conjunto de entrenamiento. Un conjunto de entrenamiento es una serie de formas correctamente clasificadas mediante visión humana que se introducen al sistema de visión previamente a su funcionamiento. El sistema de visión irá utilizando estas muestras para la clasificación de las nuevas formas.

### 2.6.1.1. MÉTODOS MATCHING

Las técnicas de reconocimiento basadas en matching representan cada clase mediante un vector de formas prototipo. Un patrón desconocido es asignado a la clase a la que más se aproxime en términos de una métrica predefinida. El método de clasificación más simple es el de la distancia mínima, que, como su propio nombre indica, calcula la distancia entre lo desconocido y cada vector de formas prototipo. Este método elige la distancia más pequeña para tomar una decisión. También existe una aproximación basada en la correlación, que puede ser formulada directamente en términos de imágenes y además es bastante intuitivo.

- Clasificación por mínima distancia

Para clasificar una nueva forma que se presenta al sistema de reconocimiento se calcula la distancia a los prototipos de cada clase y se asigna la forma a la clase cuyo prototipo presenta una menor distancia. Esta forma de clasificar, se puede reinterpretar como separadores entre las clases en forma de rectas, planos o hiperplanos.

El prototipo lo podemos calcular por el procedimiento de la media, en el cual se pueden excluir los elementos del conjunto de entrenamiento más sospechosos o diferentes, es decir, los elementos discordantes.

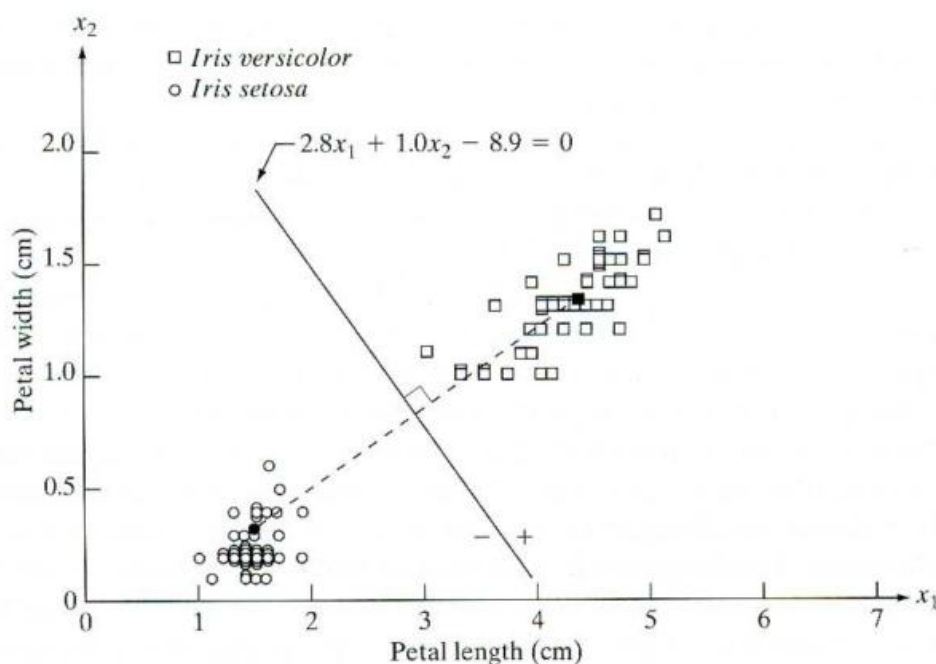




Figura 2.6.1.1.1. Ejemplo de clasificación por prototipo

En la figura 2.6.1.1.1 podemos observar un ejemplo de decisión de las fronteras del clasificador de distancia mínima para dos clases distintas de iris.

- Clasificación por el vecino más próximo

Este método solo es aplicable cuando disponemos previamente de un conjunto de entrenamiento clasificado. Se caracteriza por asignar una forma a la clase a la cual pertenece el elemento más próximo del conjunto de entrenamiento, lo que puede llegar a causar problemas en las zonas de transición entre clases o cuando existen elementos discordantes.

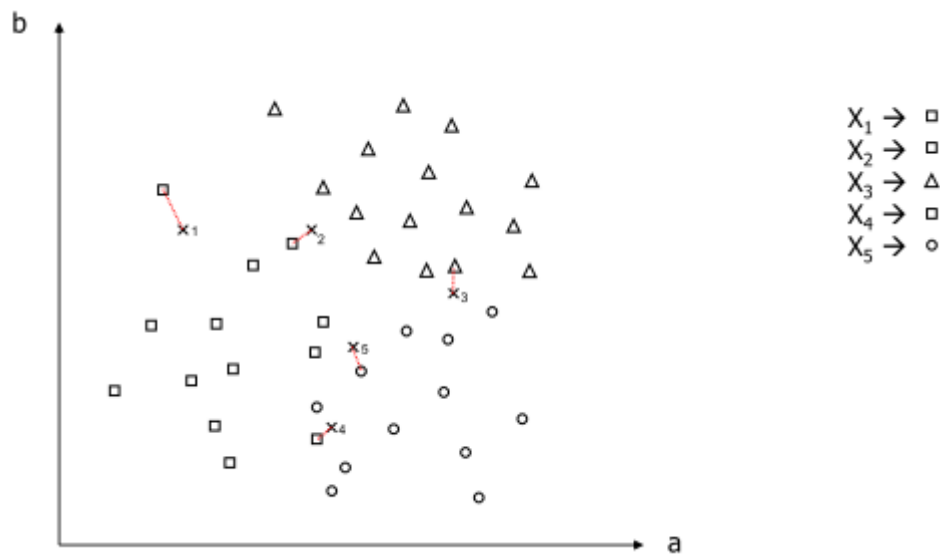


Figura 2.6.1.1.2. Ejemplo de asignación por el vecino más próximo

- Clasificación por k-vecinos

Igual que en el caso anterior este tipo de clasificación sólo se puede utilizar si previamente se dispone de un conjunto de entrenamiento clasificado. Esta forma de clasificación se basa en asignar una forma a la clase más abundante de sus k-vecinos. K es un número de vecinos que decidimos previamente y que suele ser impar para evitar empates. Este método de clasificación soluciona bastante bien el problema de los elementos discordantes del conjunto de entrenamiento si k es lo suficientemente grande.

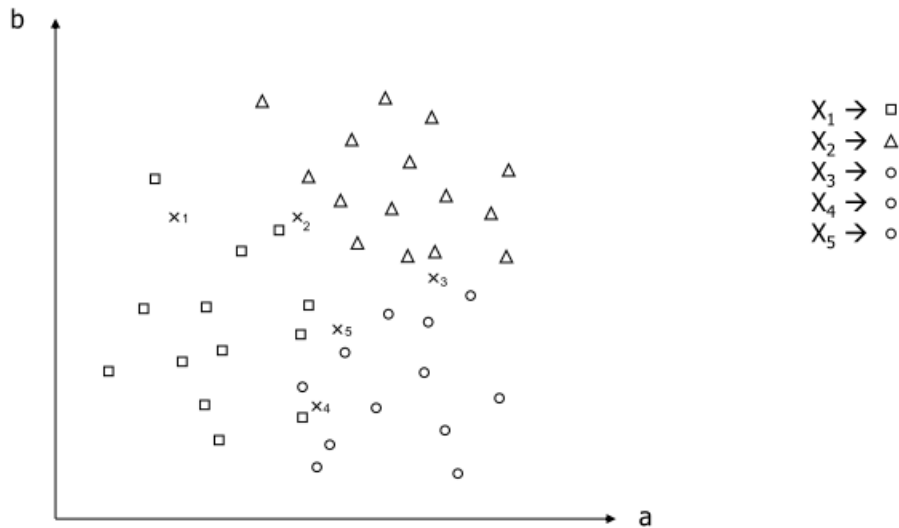


Figura 2.6.1.1.3. Ejemplo de clasificación por k-vecinos.

- Clasificación por correlación

Este método es usado cuando el vector de características de la forma es muy largo o no está clara la forma más adecuada para clasificarlo. La correlación indica la fuerza y la dirección de una relación lineal y proporcionalidad entre dos variables estadísticas. En nuestro caso, efectuaremos la correlación de la forma a clasificar con todos sus prototipos almacenados, y la que nos proporcione el pico más alto, será la más semejante.

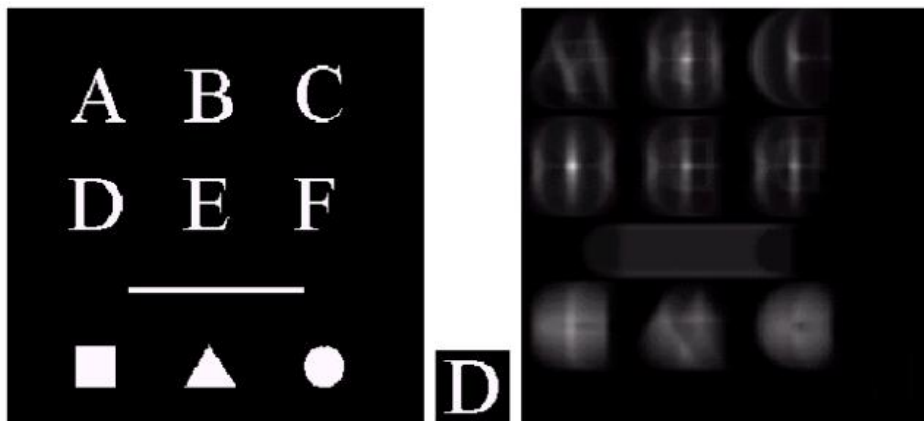


Figura 2.6.1.1.4. Ejemplo de correlación.

En la figura 2.6.1.1.4 podemos observar como al correlar la letra “D” con la imagen, produce el máximo brillo al coincidir con la misma letra.

### 2.6.1.2. REDES NEURONALES

Las redes neuronales son un procesamiento automático inspirado en la forma que funciona el sistema nervioso de los animales. Se trata de un sistema de interconexión de neuronas en una red que colabora para producir un estímulo de salida. Están íntimamente relacionadas con la inteligencia artificial.

Son sistemas de clasificación masivamente paralelo formado por neuronas conectadas entre sí. Las entradas se denominan estímulos y su salida el resultado de la clasificación. Estos estímulos son medidas y parámetros de los objetos y el tipo de neurona más comúnmente utilizado es el perceptrón.

El perceptrón es un sistema que posee un conjunto de entradas que se multiplican por unos coeficientes llamados pesos, estos coeficientes posteriormente serán sumados. La salida puede ser binaria o bien con un comportamiento difuso. Por otro lado la neurona nos permite separar en dos clases según el resultado de salida.

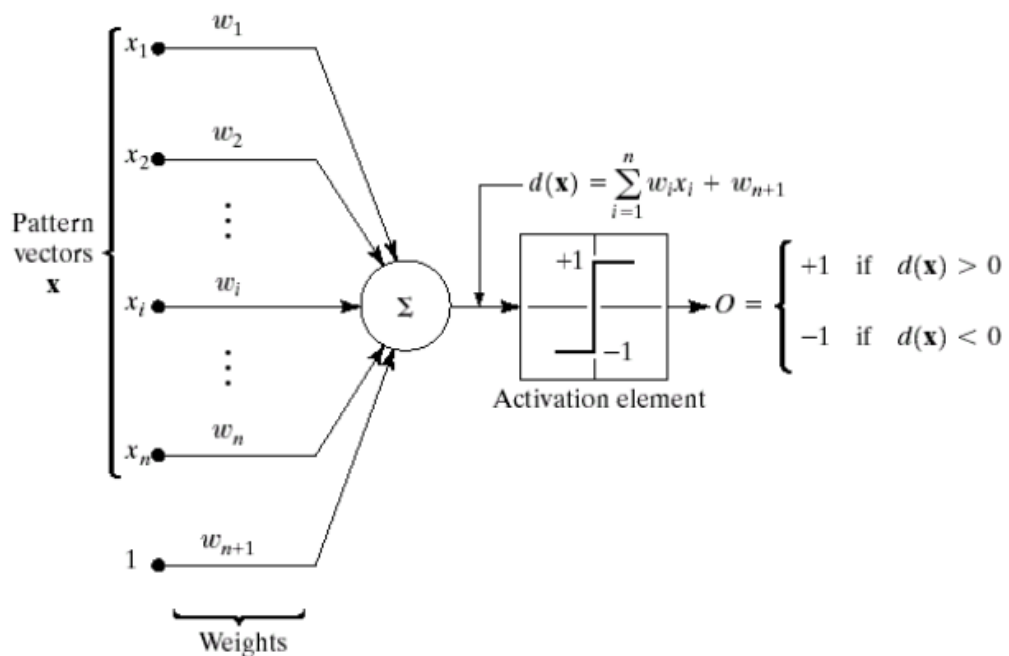


Figura 2.6.1.2.1. Funcionamiento del perceptrón.

### **2.6.2. MÉTODOS ESTRUCTURALES**

Son métodos que aplicaremos cuando los descriptores de los objetos concuerden con los descriptores relacionales. Su principal aplicación es el reconocimiento de voz por lo que en nuestro caso no los trataremos.

Entre los métodos estructurales destacan los HMM (Hidden markov Models, Modelos Ocultos de Markov), modelos estadísticos en los que se asume que el sistema a modelar es un proceso de Markov de parámetros desconocidos. El objetivo es determinar los parámetros desconocidos de dicha cadena a partir de los parámetros observables. Se basan en máquinas de estados donde se modelan las transiciones entre estados.

---

# **Parte 3:**

# **Implementación del**

# **algoritmo**

---

### 3. IMPLEMENTACIÓN DE ALGORITMOS PARA EL PROCESADO

A continuación hablaremos del algoritmo que hemos implementado para llevar a cabo la detección de humo automática del programa. Para ello emplearemos muchas de las funciones que hemos nombrado anteriormente en la introducción teórica, como las operaciones morfológicas de erode y dilate, o el histograma de la imagen en movimiento.

Para ello llevaremos a cabo principalmente 5 pasos:

- Adquisición de las imágenes: aquí nos encargaremos de adquirir frame a frame todos los frames del video que vayamos a usar. Sobre estas imágenes realizaremos la detección de humo.
- Detección de movimiento: en esta etapa detectaremos si se ha producido movimiento en el video o no. Este movimiento no tiene porqué ser humo, sino cualquier clase de movimiento. Lo calcularemos mediante la resta de un frame y su anterior.
- Eliminación de ruido: donde nos encargaremos de eliminar todos los pequeños puntos resultantes de la detección de humo que no nos interesen y nos puedan llevar a algún error. Aquí será muy conveniente emplear operaciones morfológicas como erode y dilate que veremos más adelante.
- Propiedades: etapa donde obtendremos las distintas propiedades de la imagen que nos sean necesarias para la detección de humo.
- Detección de humo: etapa final donde remarcaremos lo que se trate de humo realmente y no movimiento.

## 3.1. INTRODUCCIÓN A LA HERRAMIENTA DE TRABAJO GUIDE

En este apartado nos encargaremos de explicar la herramienta de interfaz gráfica con la que hemos creado nuestro programa.

GUIDE es un entorno de programación visual disponible en MATLAB para realizar y ejecutar programas que necesiten ingreso continuo de datos. Tiene las características básicas de todos los programas visuales como Visual Basic o VisualC++.

Una aplicación GUIDE consta de dos archivos: .m y .fig. El archivo .m es el que contiene el código con las correspondencias de los botones de control de la interfaz y el archivo .fig contiene los elementos gráficos.

Cada vez que añadamos un nuevo elemento en la interfaz gráfica, se generará automáticamente el código en el archivo .m.

### 3.1.1. MANEJO DE DATOS ENTRE LOS ELEMENTOS DE LA APLICACIÓN Y EL ARCHIVO .M

Todos los valores de las propiedades de los elementos (color, valor, posición, string...) y los valores de las variables transitorias del programa se almacenan en una estructura. Para acceder a ellos se hace mediante un único y mismo identificador para todos.

“Handles” es nuestro identificador a los datos de la aplicación y `guidata` es la sentencia para guardar los datos de la aplicación.

`Guidata` es la función que guarda las variables y propiedades de los elementos en la estructura de datos de la aplicación, por lo tanto, como regla general, en cada subrutina se debe escribir en la última línea lo siguiente:

```
➔ guidata(hObject,handles);
```

Esta sentencia nos garantiza que cualquier cambio o asignación de propiedades o variables quede almacenado. Por ejemplo, si dentro de una subrutina una operación dio como resultado una variable `x` para poder utilizarla desde el programa u otra subrutina debemos guardarla de la siguiente manera:

```
➔ handles.x=x;  
➔ guidata(hObject,handles);
```

La primera línea crea la variable `handles` a la estructura de datos de la aplicación apuntada por `handles` y la segunda graba el valor.

### 3.1.2. SENTENCIAS GET Y SET

La asignación u obtención de valores de los componentes se realiza mediante las sentencias `get` y `set`. Por ejemplo, si queremos que la variable `x` tenga el valor de una casilla con Tag: `Hola`, tendremos que escribir:

- `x= get(handles.Hola,'Value');`

Para asignar el valor a la variable `x` al `statictext` etiquetada como `texto` escribimos:

- `set(handles.texto,'String',x);`

Con esta instrucción asignaremos el valor de la variable `x`, al `statictext` llamado `texto`.

### 3.1.3. INICIANDO GUIDE

Primer paso para abrir e iniciar la creación de nuestro programa con GUIDE:

- File → New → GUI

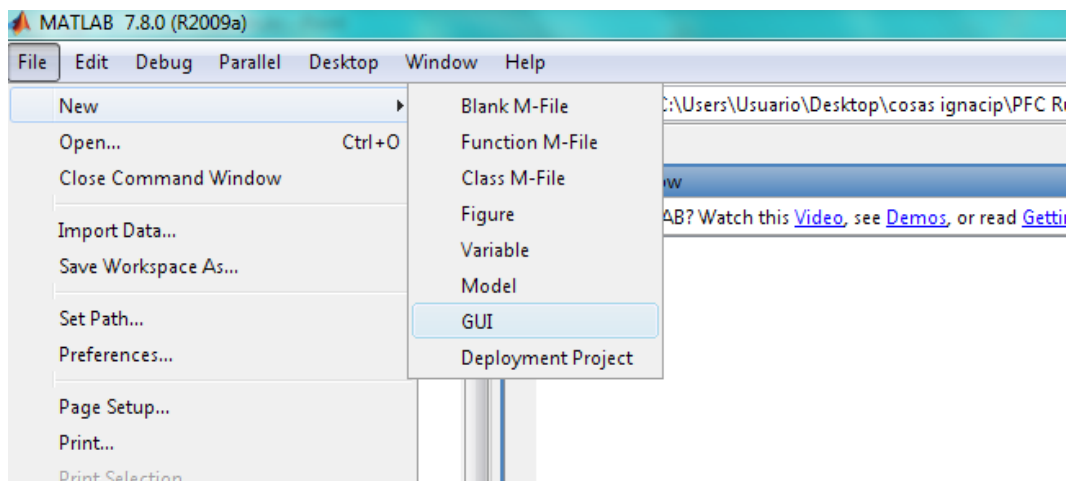


Figure 3.1.3.1. Abrir un GUI con MatLab

Tras ejecutar este paso, se abrirá la ventana para crear nuestro programa en MatLab, donde veremos que el menú está organizado de la siguiente manera:



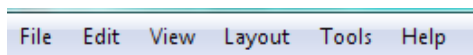
- Una barra de herramientas en la parte superior del programa con las opciones que más utilizaremos para la creación de nuestro programa.



- Una barra en el lateral izquierdo para la creación de los botones, gráficas, tablas, etc; que necesitemos.



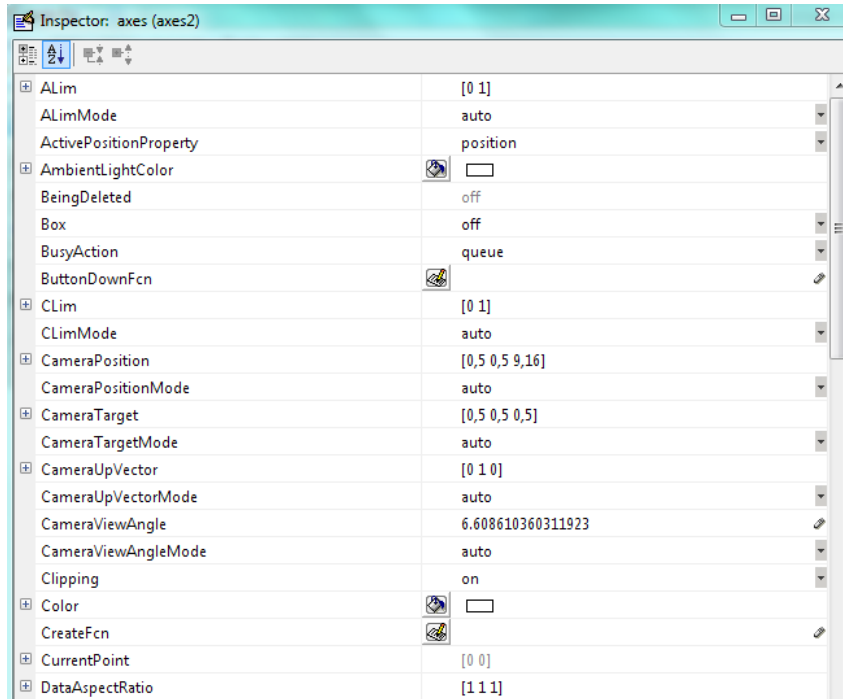
- Un menú en la parte superior con las opciones de File, Edit, View, Layout, Tools y Help.



### 3.1.4. PARTES DE GUIDE

A continuación haremos un breve repaso a las partes de GUIDE más comunes y que más utilizaremos a lo largo de la creación de nuestro programa:

- Inspector de propiedades: es una ventana que abriremos haciendo doble click sobre el objeto en cuestión que queramos cambiar las propiedades. Aquí podremos cambiar el color, la fuente, el tamaño, etc.



Dentro de esta ventana debemos destacar las principales propiedades que en nuestro caso modificaremos:

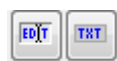
- Background Color: Cambia el color de fondo del control que hayamos seleccionado
  - Callback: Probablemente la propiedad más importante de este panel. Con ella le diremos al control que hacer cuando éste se active.
  - Enable: Activa o desactiva dicho control.
  - Tag: Con esta propiedad podremos identificar el control para que a la hora de crear el fichero .m nos sea más sencillo saber qué control es el que estamos programando.
- Run figure: para poder guardar nuestro programa y poder visualizar y probar el GUI.



- Push Button: crea el típico botón, que al pulsarlo ejecutara una acción determinada.



- Edit text y Static text: crearemos un cuadro de texto, que en el caso de Edit text podremos editar durante la ejecución del programa.



- Axes: sirve para crear áreas ya sea para mostrar gráficos, imágenes o videos.



- Panel: crea un marco donde poder agrupar diferentes controles.



## 3.2. ENTORNO GRÁFICO DE TRABAJO

En esta sección nos encargaremos de mostrar el entorno gráfico de nuestro programa y explicar cada una de las partes detalladamente.

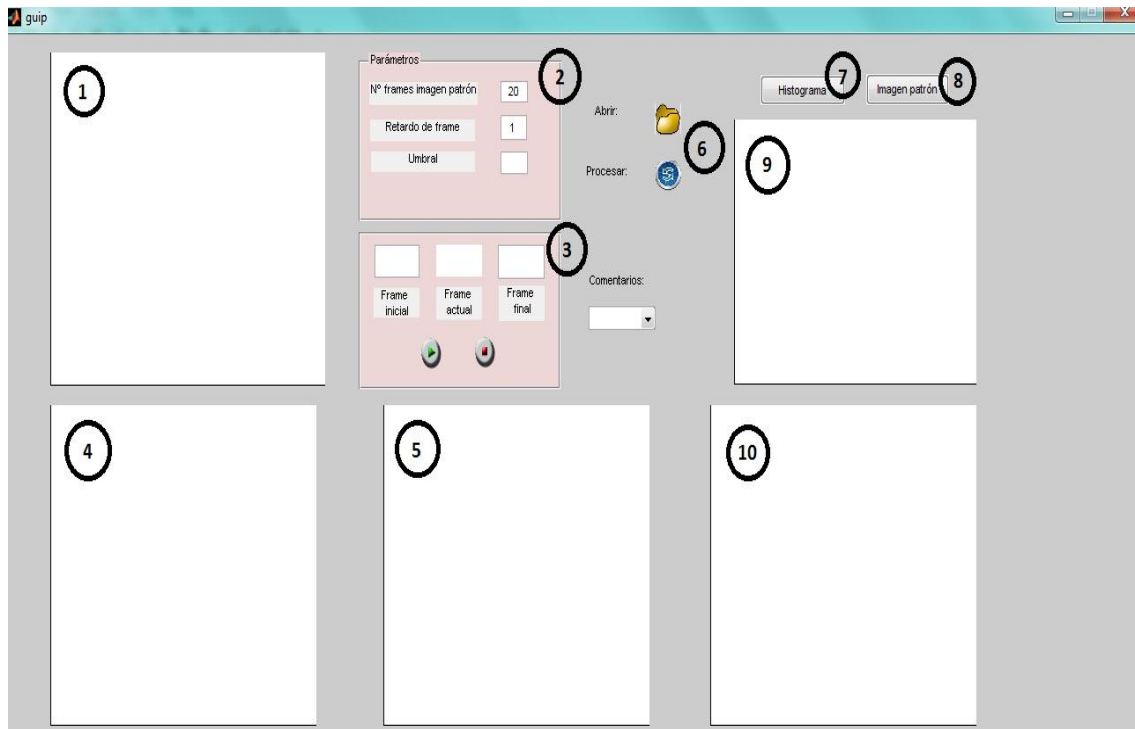


Figura 3.2.1. Entorno gráfico de trabajo

1. En esta ventana mostraremos el video original. Aquí lo reproduciremos ya sea frame a frame como el original o a saltos que definirá el usuario.
2. Esta es la ventana de parámetros. Consta de tres parámetros editables: número de frame de la imagen patrón, retardo de frame y umbral.

El número de frame de la imagen patrón será por defecto 20, a no ser que el usuario defina editarlo. Esta imagen patrón servirá para comenzar la comparación de los distintos frames y detectar si existe humo en el video o no.

Retardo de frame: este parámetro será por defecto 1. Lo podremos editar definiendo con esto los saltos que hará el video, es decir, si el video se procesará frame a frame, si irá a saltos de dos, cinco, o los que decida el usuario.

Umbral: por defecto el umbral será calculado frame a frame de forma automática por el programa asignando el umbral más óptimo para cada frame con la función:

→umbral =graythresh(imagen);

Si en cambio el usuario decide fijar el umbral, éste permanecerá constante durante todo el procesado.

3. En esta ventana tenemos los parámetros propios del video que carguemos. El frame inicial por defecto será 1, aunque tendremos la posibilidad de editarlo si el usuario lo desea. El frame final por defecto será el último frame del video que carguemos, aunque al igual que el frame inicial, lo podremos modificar a nuestro antojo con el valor que más nos convenga. Por último, el frame actual será una ventana informativa que no podremos editar; y que nos irá informando del frame en el que nos encontramos durante el procesado del video. También tenemos una tecla de reproducción, con la que reproduciremos el video original sin hacer ningún procesado en la ventana 1; y una tecla de stop para parar bien la reproducción del video o bien el procesado que se esté realizando.

4. En esta ventana se irá mostrando durante el procesado la resta de un frame con el anterior en binario. Esta imagen será la que obtendremos sin haberle aplicado ningún tipo de operación, simplemente la diferencia de ambos frames.

5. Aquí se mostrará la imagen mostrada en la ventana 4, tras haberle aplicado en primer lugar las operaciones morfológicas de erode y dilate, es decir close. Con esta operación eliminaremos la mayoría de los pequeños puntos que se muestran en la imagen que no nos convienen a la hora de la detección de humo, ya que pueden llegar a provocar problemas.

Con esta operación solo no llegaremos a eliminar todo lo que nos molesta, por lo tanto tendremos que aplicar a continuación una operación que elimine los objetos con área menor a una establecida por el usuario (área mínima). El resultado de estas dos operaciones es el que mostramos en la ventana 5.

6. Abrir: pulsando este botón se nos abrirá el directorio donde elegiremos el video que queramos procesar.

Procesar: pulsándolo empezaremos el procesado.

7. Histograma: calcula y muestra en la ventana 9 el histograma del video que vamos a procesar.  
La función en MatLab que utilizaremos será:  
  
→`histograma=imhist(imagen);`
8. Imagen patrón: muestra en la ventana 9 el número de imagen patrón que le hemos indicado antes al programa, y si no, mostrará por defecto el frame 20.
9. Ventana que como anteriormente hemos indicado, mostrará dependiendo de lo que se le indique, el histograma o la imagen patrón del procesado.

### 3.3. ALGORITMO PARA EL PROCESADO

Dentro de nuestro programa el bucle principal que se encargará de realizar el procesado será un while. A este bucle entraremos siempre y cuando se cumpla la siguiente condición:

- El frame en el que nos encontremos deberá ser igual o menor al número total de frames menos el número que hemos como frame patrón. El motivo por el que restamos a la condición el número de frame de la imagen patrón es porque si no el programa llegará a un punto que nos de error por haber llegado a un frame que ya no existe.

Si esta condición no se cumple el programa se parará, limpiará todas las variables y estará preparado para un nuevo procesado.

Una vez estemos dentro del bucle, empezaremos con el procesamiento. En primer lugar, calcularemos a partir de la imagen patrón y su siguiente frame, la resta de ambas imágenes para poder visualizar lo que ha cambiado de una a otra. Para poder trabajar con el resultado de la resta deberemos convertir dicha imagen en una imagen binaria, es decir, una imagen negra donde lo que este blanco sea el resultado de la resta. Esta imagen resultante la podremos observar en una de las ventanas de nuestro entorno de trabajo.

A continuación, será necesario realizar un bucle que se encargue de decidir en qué momento será necesario calcular de forma automática el umbral o dejarlo constante tal y como el usuario había indicado. El bucle será de la siguiente manera:

- Si estamos en la primera pasada del programa, éste detectará si el usuario a escrito un valor en la casilla umbral o no. Si lo ha escrito, el valor permanecerá constante durante todo el proceso; en cambio, si el usuario no lo ha escrito, cada pasada calcularemos un umbral óptimo de forma automática con la función “grayhresh”.

El siguiente paso será eliminar todos los pequeños objetos no deseados que hayan quedado en la imagen tras la resta. Los eliminaremos con la operación open: erode+dilate. Esta operación la realizaremos en forma de disco para conseguir imitar la forma redondeada del humo.

Al hacer la operación anterior podemos observar en la imagen que no hemos conseguido eliminar todos los pequeños objetos que nos molestaban para llevar a cabo la detección del humo. Por lo tanto tendremos que realizar una segunda operación de eliminación de objetos de la siguiente manera:

- Fijaremos una área mínima que nos sirva de base para decirle al programa que todo aquel objeto que sobrepase dicha área sea eliminado. Al igual que los demás parámetros, éste podrá ser fijado por el usuario según su conveniencia. Esta imagen también será mostrada en nuestro entorno de trabajo para que el usuario pueda comprender los pasos que se están llevando a cabo.

Por último contornearemos la región que hemos detectado como humo con la función “imcontour”, y mostraremos el resultado en la ventana correspondiente.

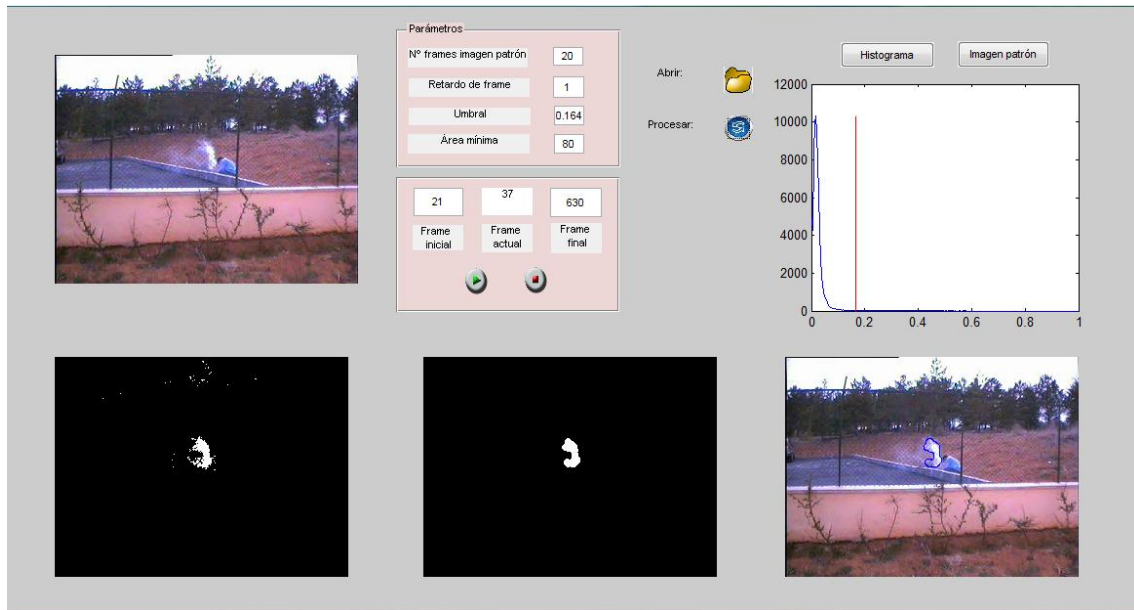
Este algoritmo al basarse en la diferencia de un frame con otro, solo podremos utilizarlo con videos donde la cámara permanezca en una posición fija.



### 3.3.1. EJEMPLOS DEL PROCESADO

En este apartado vamos a visualizar algunos ejemplos resultantes de la simulación.

- Simulación con el primer video:

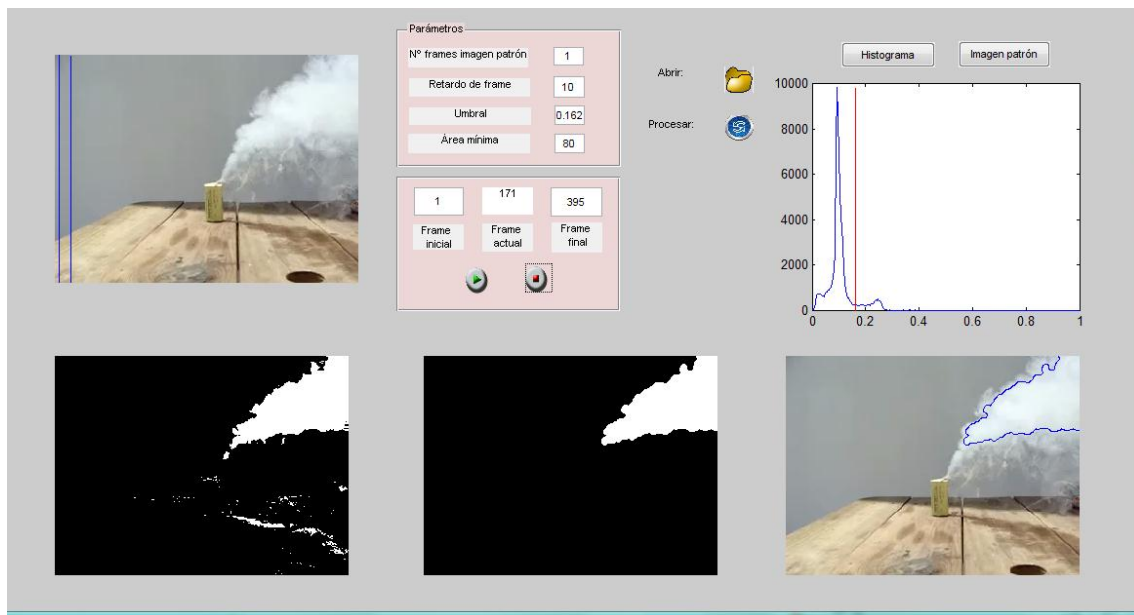


En la figura anterior podemos observar los pasos que hemos ido nombrando anteriormente. En la ventana inferior izquierda tenemos la resta de frames, en la cual podemos ver como hay objetos que nos molestan para realizar el procesamiento y que son eliminados en la siguiente figura, llegando al resultado final donde el humo es contorneado y detectado.



Como podemos observar, los resultados obtenidos son bastantes buenos. Un entorno donde no hay mucho movimiento y además donde el fondo contrasta bastante con el humo, nos ha permitido realizar una buena detección de humo.

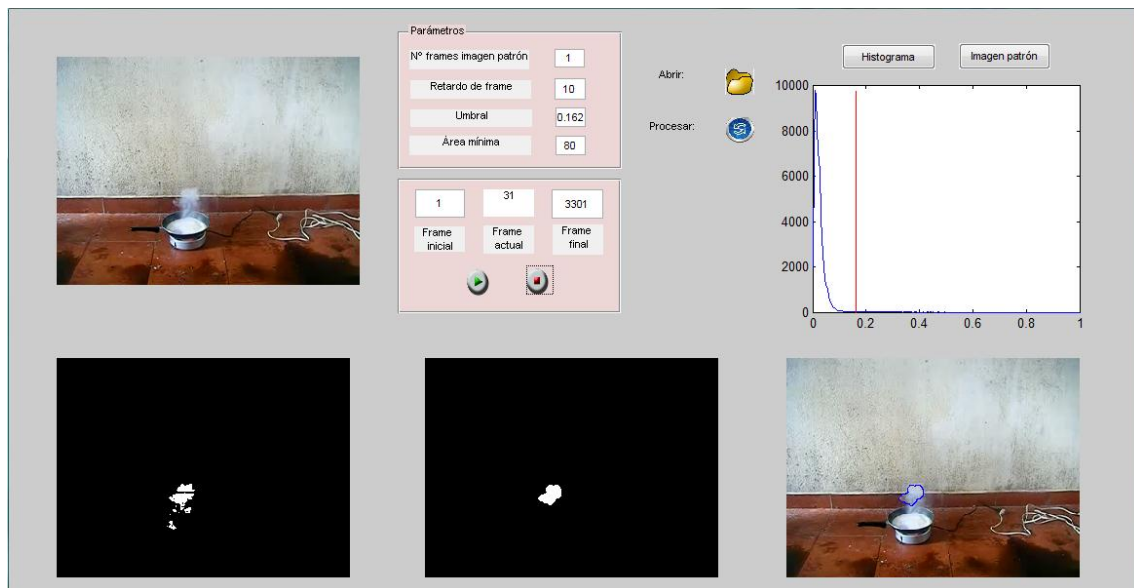
- Simulación con el segundo video:



Esta simulación no es tan buena como la anterior, ya que tiene varias dificultades añadidas. Se trata de un video donde el fondo y la mesa son colores claritos y el contraste con el humo es demasiado pequeño. Además como podemos observar en

la imagen, el humo produce una sombra en la mesa que causa muchos problemas y que debemos evitar.

- Simulación con el tercer video:



Por último, la detección de este tercer video ha sido complicada porque al igual que la simulación anterior, tenemos un fondo muy parecido al humo, con lo que es muy

difícil poder detectar si ahí hay humo o no. Además la cantidad de humo es pequeña con lo que no podemos eliminar muchas manchas que nos molestan porque a su vez estamos eliminando el humo.

## 4. CONCLUSIONES Y LINEAS DE FUTURO

Este proyecto surgió por parte de mi tutor Ignacio Bosh con la idea de mejorar un proyecto que anteriormente se había llevado a cabo pero que había quedado bastante inacabado. Por lo tanto hemos intentado mejorarlo, sobre todo en la parte visual para lograr un mayor entendimiento del proceso que estaba siendo llevado a cabo.

Hemos hecho del programa, un programa mucho más intuitivo, el cual pueda llegar a utilizar cualquier usuario sin muchos conceptos previos, empleando entre otros más ventanas explicativas.

Otro de los objetivos ha sido generalizar el programa para que este pueda ser usado con cualquier video donde aparezca humo y esté grabado con la cámara fija. Para ello hemos hecho de los parámetros que anteriormente estaban fijados para un video en concreto, parámetros que el usuario pueda modificar a su antojo y adaptarlos al video que quiera analizar.

Gracias a este proyecto de fin de carrera, he conseguido familiarizarme más con el programa MatLab y sobretodo empezar de cero con la herramienta para crear interfaces gráficas. Además he reforzado y sentado todos los conocimientos sobre el tratamiento digital de imágenes.

En conclusión, todos los objetivos que nos marcamos en un primer momento, hemos conseguido llevarlos a cabo en la medida de lo posible, aunque el tiempo para realizarlos no haya sido todo el que nos hubiera gustado.

Como líneas de futuro, cabe mencionar que el programa pueda seguir siendo mejorado, en el aspecto del algoritmo sobretodo, para conseguir mejores resultados, y mejorar el rendimiento del programa.

Unas de las aplicaciones reales que podría tener este programa podría ser su aplicación en una webcam junto con un detector de humos convencional, para lograr tener dos recursos a la hora de detectar humos. Es decir, tener como una confirmación o segunda opinión a la hora de detectar un posible incendio. Además en puntos donde hay riesgo de incendio, instalando una cámara web emitiendo en directo con este programa, podría ayudarnos a combatir muchos de los incendios que cada año son producidos.

## 5. BIBLIOGRAFÍA

- Rafael C. Gonzalez y Richard E. Woods. Digital Image Processing, Second Edition.
- William K. Pratt. Digital Image Processing, Fourth Edition.
- Rafael C. Gonzalez y Richard E. Woods. Digital Image Processing Using MatLab.
- John C. Russ. The Image Processing Handbook, Sixth Edition.
- Alegre E. , Sánchez L., Fernández R. A., Mostaza J. C. Procesamiento Digital de Imagen: fundamentos y prácticas con MatLab, Universidad de León.
- The MathWorks Image Processing Toolbox, for use MatLab 2006.
- [www.youtube.com](http://www.youtube.com)
- [www.mathworks.es](http://www.mathworks.es)
- [www.wikipedia.com](http://www.wikipedia.com)
- Apuntes del curso 2011/2012 asignatura: Tratamiento Digital de Imágenes.
- Otros Trabajos de Fin de Carrera.

## 6. ÍNDICE DE FIGURAS

### PARTE 1:

- 1.1.1. Restauración de imágenes
- 1.1.2. Estructura de una red neuronal de reconocimiento de caracteres
- 1.1.3. Imagen original e imagen comprimida
  
- 1.2.1. (a) Imagen original; (b) Imagen muestreada cada 2 píxeles;  
(c) Imagen muestreada cada 4 píxeles; (d) Imagen muestreada cada 8 píxeles
  
- 1.3.1. Esquema de procesamiento digital de imágenes

### PARTE 2:

- 2.1.2.1. Muestreo de una señal
- 2.1.2.2. Señal codificada con 8 niveles, es decir, con 3 bits
- 2.1.3.1. Ejemplo de 4-conectividad
- 2.1.3.2. Ejemplo de 8-conectividad.
- 2.1.3.3. Ejemplo de conectividades.
  
- 2.2.1.1.1. Ejemplo transformación negativo.
- 2.2.1.1.2. Ejemplo de modificación de brillo de una imagen.
- 2.2.1.1.3. Ejemplo de modificación de contraste de menos a más.
  
- 2.2.1.2.1. Histogramas más comunes.
  
- 2.2.1.3.1. Ejemplo de sustracción de imágenes.
  
- 2.3.1.4.1 Ejemplo del promediado de imágenes en astronomía.
  
- 2.2.2.1.1. Aplicación filtro de media.
- 2.2.2.1.2. Ejemplos de máscaras Gaussianas.
- 2.2.2.2.1. Ejemplo de filtro de realce.
  
- 2.2.3.1. Pasos básicos para el filtrado en el dominio de la frecuencia.

2.2.3.2. Ejemplo de filtro paso bajo o de suavizado en el dominio de la frecuencia.

2.2.3.3. Ejemplo de filtro paso alto o de realce en el dominio de la frecuencia.

2.3.1. Restauración de imágenes

2.3.2. Imagen restaurada.

2.4.1.1. Ejemplo de histograma donde se utiliza un umbral o varios.

2.4.2.1. Ejemplo de utilización del método crecimiento de regiones.

2.5.1.2. Ejemplo del método partir y unir.

2.4.3.1. Ejemplo máscara.

2.4.3.2. Ejemplo máscara 2

2.4.3.3. Gradiente de una imagen.

2.5.1.1. Numeración de dirección para 4-conectividad y 8-conectividad.

2.5.1.2. Ejemplo de cálculo del número de forma.

2.5.1.3. Ejemplo de aproximación poligonal.

2.5.1.4. Ejemplo de esqueleto de figuras.

2.5.2.1. Fórmula de la compacidad.

2.5.2.2. Ejemplo de cálculo de elongación y excentricidad.

2.5.3.1. Ejemplo de descriptores relacionales.

2.6.1.1.1. Ejemplo de clasificación por prototipo.

2.6.1.1.2. Ejemplo de asignación por el vecino más próximo.

2.6.1.1.3. Ejemplo de clasificación por k-vecinos.

2.6.1.1.4. Ejemplo de correlación.

2.6.1.2.1. Funcionamiento del perceptrón.

## **PARTE 3:**

3.1.3.1. Abrir un GUI con MatLab

3.1.4.1. Inspector GUI



### 3.2.1. Entorno gráfico de trabajo