

Procesamiento de Imágenes

Estrategia: Es una materia muy teórica en los exámenes por lo que una guía de estudio con la mayor información posible vista en el parcial, será lo mejor. Las tareas pueden rescatar puntos así como condenar la calificación

▼ Primer Parcial

▼ 09-08-2021

Profesor: Luis Fernando Gutiérrez Marfileño

Para la programación del curso usaremos Java, C# o MATLAB

EVALUACIÓN DE LOS APRENDIZAJES

- Participación (que consta principalmente de aportación de ideas y respuestas a preguntas directas en clases): 10%
- Exposición (que consiste de una plática frente a grupo, por parte del alumno, sobre algún tema a investigar): 10%
- Implementación de algoritmos (que consiste de los programas que se entreguen, como resultado de la implementación de modelos aprendidos): 30%
- Exámenes 20%
- Proyecto final (que consiste en realizar alguna aplicación sobre lo aprendido durante el curso): 30%

12

Proyecto:

Contenido: El proyecto final deberán incluir todos los elementos señalados en el documento respectivo, si falta alguno de ellos se reflejará en la calificación asignada

Fecha de entrega: Se señalará en el documento respectivo, si no se entrega en la fecha señalada ya no se recibirá

Calificación: Si el proyecto se entrega en tiempo y forma tendrá una calificación de 10, si no, se considerarán los elementos faltantes y afectarán a la calificación.

Ponderación: El proyecto representa un 30% de la calificación final

▼ 10-08-2021

Tema 1.1: Concepto de imagen

Imagen: Reproducción de la figura de un objeto por la combinación de los rayos de luz que proceden de él

Normalmente pensamos en una imagen en el sentido de una representación plana del brillo, es decir, la cantidad de luz reflejada o transmitida por un objeto

Matemáticamente una imagen suele ser una función de dos variables espaciales, por ejemplo $f[x, y]$ que representa el brillo f en la ubicación cartesiana $[x, y]$

También puede ser graficado en tres dimensiones con el brillo mostrado en el eje z

Es cada vez más común tratar con imágenes que tienen más de dos dimensiones coordinadas, por ejemplo:

- $f[x, y, t_n]$ "película" monocromática, conjunto discreto de imágenes a lo largo del tiempo
- $f[x, y, \lambda]$ imagen espectral con dominio continuo de longitudes de onda
- $f[x, y, \lambda_n]$ imagen multiespectral, conjunto discreto de longitudes de onda
- $f[x, y, t]$ imagen monocromática variable en el tiempo en un dominio de tiempo continuo
- $f[x, y, t_n]$ imagen monocromática variable en el tiempo con muestras de tiempo discretas (cin)
- $f[x, y, z]$ imagen monocromática 3D (p. ej., holograma óptico)
- $f[x, y, t_n, \lambda_m]$ muestras discretas en tiempo y longitud de onda, p. ej., película en color
- $f[x, y, z, t, \lambda]$ realidad

Por lo general, es posible cortar porciones bidimensionales de estas funciones multidimensionales para crear imágenes, pero las imágenes no tienen por qué ser pictóricas (pictórico es como un pintor que pinte algo realista)

Por ejemplo, considerar el 2D rebanadas cortadas de la función 3D función espacio-temporal $f[x, y, t]$ la rebanada 2D $f[x, y, t; t = t_0]$ es pictórico pero $f[x, y = y_0, t]$ no lo es

Después de convertir la información de la imagen en una matriz de números enteros, la imagen puede ser manipulada, procesada y mostrada por computadora.

El procesamiento informático se utiliza para la mejora de imágenes, restauración, segmentación, descripción, reconocimiento, codificación, reconstrucción o transformación

- En esta función la amplitud de f en cualquier par de coordenadas (x, y) se llama **intensidad** o **nivel de gris** de la imagen en ese punto
- Cuando x, y , y los valores de intensidad de f son todas cantidades finitas y discretas, se le llama **imagen digital**
- El campo del **procesamiento de imágenes digitales** se refiere al procesamiento imágenes digitales por medio de una computadora digital
- Se debe tener en cuenta que una imagen digital se compone de un número finito de elementos, cada uno de los cuales tiene una ubicación y un valor particulares

- Estos elementos se denominan elementos de imagen, elementos de imagen, pels y píxeles
- **Pixel** es el término que se usa más ampliamente para denotar los elementos de una imagen digital

▼ 11-08-2021

- **EL PROCESAMIENTO de imágenes** tiene como objetivo mejorar el aspecto de las imágenes y hacer más evidentes en ellas ciertos detalles que se desean hacer notar
- La imagen puede haber sido generada de muchas maneras, por ejemplo, fotográficamente, o electrónicamente, por medio de monitores de televisión

Se puede hacer, en general, por medio de métodos ópticos, o bien por métodos digitales en una computadora

Procesamiento óptico

Los principios del procesamiento óptico de imágenes están bien establecidos desde el siglo pasado, cuando se desarrolló la teoría de la difracción de la luz

Su aplicación data apenas de los 60's con el rayo láser

Se basa en el hecho de que la imagen de difracción de Fraunhofer de una transparencia colocada en el plano focal frontal de una lente es una distribución luminosa que representa la distribución de las frecuencias de Fourier que componen la imagen, a la que se le llama técnicamente la transformada de Fourier

Procesamiento digital

Sus fundamentos no se llevaban a cabo debido a la falta de computadoras

Con la aparición de las computadoras, era natural que se comenzara a desarrollar este campo

Uno de los primeros lugares donde se realizó el proceso digital fue en el Jet Propulsion Laboratory en 1959 para mejorar imágenes enviadas por cohetes

Los resultados fueron obtenidos en tiempo relativamente corto. Pronto se extendieron las aplicaciones del método a otros campos

Procedimiento

- El procesamiento digital de imágenes se efectúa dividiendo la imagen en un arreglo rectangular de elementos
- Cada elemento de la imagen así dividida se conoce con el nombre de pixel
- El siguiente paso es asignar un valor numérico a la luminosidad promedio de cada pixel
- Así, los valores de la luminosidad de cada pixel, con sus coordenadas que indican su posición, definen completamente la imagen
- Todos estos números se almacenan en la memoria de una computadora

Los pasos siguientes serán la representación de los números almacenados

- La utilidad del procesamiento de imágenes es muy amplia y abarca muchos campos
- Un ejemplo son las imágenes obtenidas con fines de diagnóstico médico
- Otro ejemplo son las imágenes aéreas obtenidas para realizar exámenes del terreno
- Mediante este método se pueden analizar los recursos naturales, las fallas geológicas del terreno, etcétera

Realce

- Mejora y correcciones (degradaciones: bajo contraste, ruido, desenfoque (*blur*), orientación, ciertas distorsiones, etc.), especialmente para llevar a intervalos característicos de visión humana ("visualizar")
- En principio no implica modelos del original o de la degradación (mejora empírica, sin que necesariamente coincida con imagen original o condiciones originales, que pueden no ser adecuadas: baja iluminación, o demasiada (por ejemplo una imagen del Sol, datos de ultrasonido, o en rayos infrarrojos, invisibles al ojo humano)

12

Restauración

- Puede implicar mejora y correcciones, pero además, en base a modelos (del objeto o escena original, o de la degradación sufrida), implica recuperar (restaurar) la imagen original (lo mejor posible), o alguna de sus características
- Corrección de distorsiones (geométricas Y/O del atributo): registro (alineación correcta con transformaciones, inclusive no-lineales), normalización, etc.

14

El objetivo usualmente es menos de visualización y más de recuperación y/o análisis cuantitativo y automático (a veces también se le denomina "pre-procesamiento" en el sentido de preparar la información y uniformizarla para su estudio)

Pero se puede desear solo una imagen puramente restaurada como en la obras de arte

- Criterios de fidelidad de imagen en función del grado de similitud con el original o alguna referencia, o del modelo de degradación o modelo de cómo debe ser el original
- Carácter más cuantitativo que cualitativo
- Mayor dependencia de proceso de adquisición y degradación, menor dependencia de la imagen particular

▼ 12-08-2021

Hicimos examen diagnóstico

▼ 13-08-2021

Tema 1.3 Representación digital

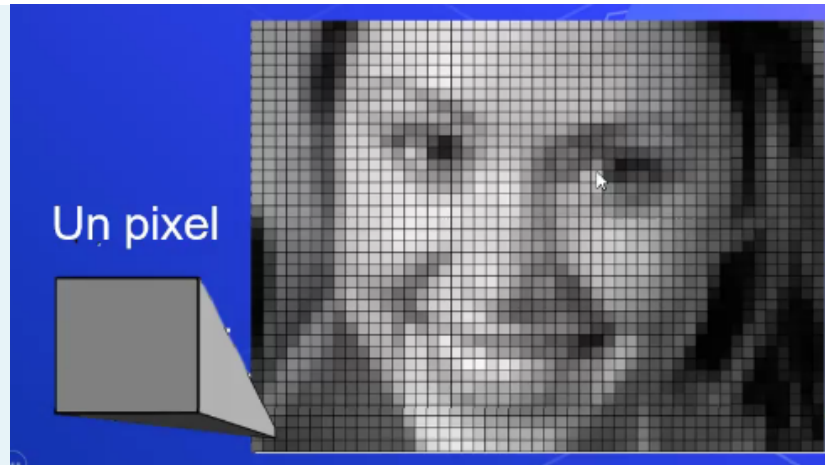
Una imagen digital es una matriz o arreglo bidimensional de números

Cada celda es un píxel

Un píxel

90	67	68	75	78	98	185	180	153	139	132	106	70	80	81	69	69	67	35	34
92	87	73	78	82	132	180	152	134	120	102	106	95	75	72	63	75	42	19	29
63	102	89	76	98	163	166	164	175	159	120	103	132	96	68	42	49	46	17	22
45	83	109	80	130	158	166	174	158	134	105	71	82	121	80	51	12	50	31	17
39	69	92	115	154	122	144	173	155	105	98	86	82	106	83	76	17	29	41	19
34	80	73	132	144	110	142	181	173	122	100	88	141	142	111	87	33	18	46	36
37	93	88	136	171	164	137	171	180	149	110	137	168	161	132	96	56	23	48	49
66	117	106	147	188	202	198	187	187	159	124	151	167	158	138	105	80	55	59	54
127	136	107	144	188	197	188	184	192	172	124	151	138	108	116	114	84	46	67	54
143	134	99	143	188	172	129	127	179	167	106	118	111	54	70	95	90	46	69	52
141	137	96	146	167	123	91	90	151	156	121	93	78	82	97	91	87	45	66	39
139	137	80	131	162	145	131	129	154	181	158	149	134	122	115	99	84	35	52	30
137	133	56	104	165	167	174	181	175	169	165	162	158	142	124	103	67	19	31	23
135	132	65	86	173	186	200	198	181	171	162	153	145	135	121	104	53	14	15	33
132	132	88	50	149	182	189	191	186	178	166	157	148	131	106	78	28	10	15	44

Una forma más común de visualizar una imagen:

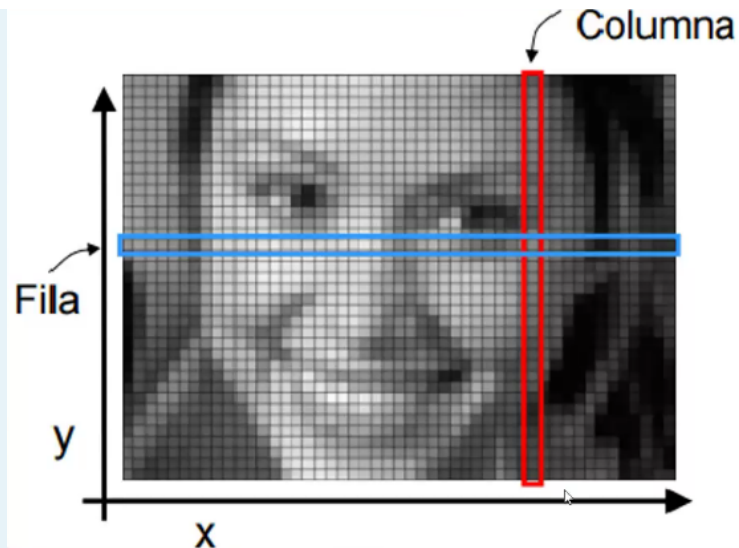


Nomenclatura

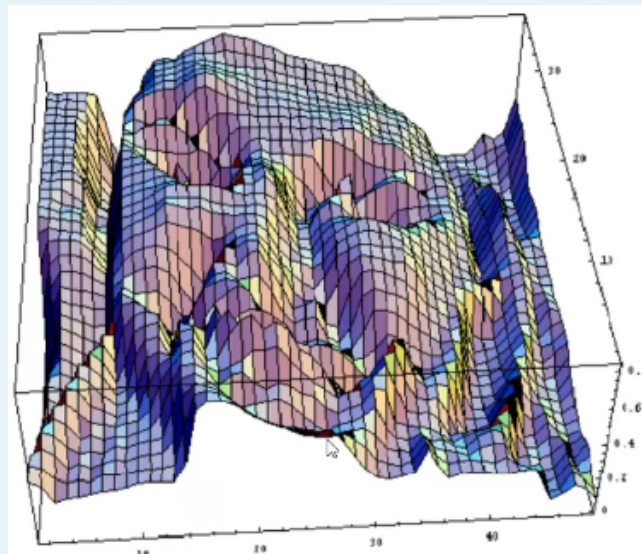
- Número de columnas de una matriz: Ancho de la imagen(width)
- Número de filas de la matriz: Alto de la imagen (height)
- Eje horizontal: Eje x
- Eje vertical: Eje y
- Normalmente el tamaño de la imagen se expresa como: Ancho x alto

Ejemplo: Tamaños típicos

- 320×240
- 640×480
- 800×600
- 1024×768



Otra posible forma de visualizar una imagen es mediante la interpretación como una superficie bidimensional. Una imagen digital es un muestreo discreto de la señal continua



Conceptos:

Vecindad en una imagen: Los píxeles próximos tienen una relación más estrecha entre sí que los lejanos. Distancia con respecto a una matriz en sentido genérico

Se espera que los valores de dos píxeles próximos sean más o menos parecidos

Tiene sentido definir la vecindad de un píxel y la distancia entre dos píxeles



Las imágenes que no tienen estas características se les conoce como "píxeleadas" ya que no manejan una vecindad cercana y es necesario tomarlo en cuenta al analizar imágenes

Supondremos un acceso indexado a los píxeles.

Si i es una imagen $i(x, y)$ será el valor del píxel en la columna x y la fila y

Cada píxel representa el valor de una magnitud física

- Cantidad de luz en un punto de escena (cantidad de radiación en la frecuencia RGB)
- Valor de color
- Nivel de radiación infrarroja, rayos X, etc, en general, cualquier radiación electromagnética
- Profundidad (distancia) de una escena en una dirección
- Cantidad de presión ejercida en un punto
- Nivel de absorción de determinada radiación

- ¿De qué tipo de datos es cada celda de la matriz?
- Imagen **binaria**: 1 píxel = 1 bit
 - 0 = negro; 1 = blanco
- Imagen en **escala de grises**: 1 píxel = 1 byte
 - Permite 256 niveles de gris
 - 0 = negro; 255 = blanco
- Imagen en **color**: 1 píxel = 3 bytes
 - Cada píxel consta de 3 valores: (Rojo, Verde, Azul)
 - Un byte por color
 - 16,7 millones de colores posibles



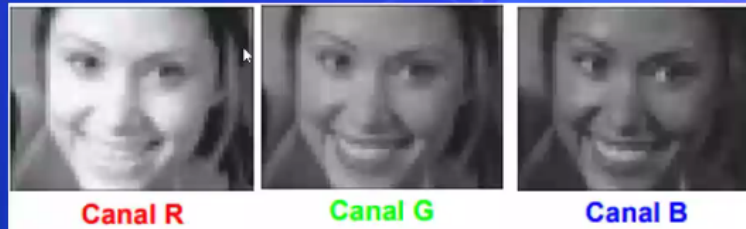
- Pero no limitado a... existen infinitos tipos posibles
- Un nivel de gris, o un color, se puede representar con más o menos bits: **profundidad** de color (*depth*)



- Hi-color: método reducido para representar colores 1 píxel = 2 bytes
 - 5 bits por cada color (Rojo, Verde, Azul)
- Imágenes en punto flotante: útiles en procesos intermedios 1 píxel = 1 float o un double

Imágenes multicanal:

- Cuando los píxeles representan magnitudes en distintos dominios físicos, decimos que la imagen es **multicanal**
- Ejemplo. Imagen en color → Imagen con 3 canales: canal R (rojo), canal G (verde), canal B (azul)



Otra forma de organizar imágenes es el RGB más el canal Alfa. El canal alfa representa el nivel de transparencia del píxel

En algunas aplicaciones suelen usarse canales para frecuencias no visibles, infrarrojo, ultravioleta, etc.



Almacenamiento de imágenes digitales

- **Cuestión 1:** ¿Cuál es el origen de coordenadas y el orden de las filas?

- Top-left: el píxel $i(0, 0)$ es la esquina superior izquierda. → Suele ser el más habitual.
- Bottom-left: el píxel $i(0, 0)$ es la esquina inferior izquierda. → Usado en algunos formatos (p.ej. BMP)

- **Cuestión 2:** ¿Cómo se almacenan los distintos canales?

- Entrelazado (*interleaved*, pixel order): $R_0, G_0, B_0, R_1, G_1, B_1, R_2, G_2, B_2, \dots, R_n, G_n, B_n$.
- No entrelazado (non-interleaved, plane order): $R_0, R_1, R_2, \dots, R_n, G_0, G_1, G_2, \dots, G_n, B_0, B_1, B_2, \dots, B_n$

BMP = Imágenes de Mapas de Bits

Resumen

- Parámetros de una imagen digital:
 - Ancho y alto
 - Número de canales y significado de cada uno
 - Número de bits por píxel y canal (*depth*)
 - Origen de coordenadas y modo de almacenamiento multicanal
- Resolución **espacial**: tamaño de la imagen
- Resolución **fotométrica**: profundidad de color
- Resolución **temporal**: aplicable en vídeos

▼ 16-08-2021

Indexación en matrices

Una matriz es un arreglo bidimensional pero puede ser n dimensional

Puede decirse que una matriz es la representación numérica almacenada de una imagen

Son elementos del mismo tipo de datos de memoria contigua

Para acceder a cada matriz es importante procesar sobre esta información

Acceder se le conoce como indexación y hay diferentes formas para acceder a un elemento específico de una matriz:

- Su posición
- En forma lineal
- En forma lógica

Lo más frecuente es especificar de manera explícita los índices de los elementos

```
A = [1 2 3 4; 5 6 7 8; 9 10 11 12; 13 14 15 16]
A = 4x4

     1     2     3     4
     5     6     7     8
     9    10    11    12
    13    14    15    16

e = A(3,2)
e = 10
```

e sería el elemento de la fila 3, columna 2

También puede hacer referencia a varios elementos al mismo tiempo especificando sus índices en un vector.

Por ejemplo, acceda al primer y tercer elemento de la segunda fila de A.

```
r = A(2,[1 3])
r = 1x2
```

5 7

Para acceder a los elementos de un intervalo de filas o columnas, utilice (:).

Por ejemplo, acceder a los elementos de la primera a la tercera fila y de la segunda a la cuarta columna de A.

```
r = A(1:3,2:4)
r = 3x3
```

2 3 4
6 7 8
10 11 12

Trabajar mucho con MATLAB

Indexar por medio de un único índice o de forma lineal

Si tenemos una matriz de 3x3 y la computadora lo guardará como una única fila, de modo que podríamos acceder desde el elemento 0 al 8

Usando valores lógicos se puede acceder sobre todo con condicionales

▼ 17-08-2021

No hubo clases

▼ 18-08-2021

Lectura de imágenes

Una imagen digital está compuesta por un código numérico mediante píxeles

Se pueden obtener por muchas formas (escáner o cámara digital)

El bit

Es la unidad de información que equivale a la elección de 0 o 1: Pasa o no pasa corriente

Bytes = 8 bits

Kilobyte = 1024 bytes

Megabyte = 1024 KB

Gigabyte = 1024 MB

Una imagen digital es una traducción o representación bidimensional de una imagen empleando bits formadas de posición e iluminación

De acuerdo en cómo son obtenidas, hay dos tipos de imágenes digitales:

- **Imágenes Digitales Procesadas (IDP)**

Las IDP resultan de lo que se conoce como procesamiento de imágenes

Son imágenes obtenidas a partir de dispositivos de conversión analógica - digital, como los escáneres o las cámaras digitales y pueden ser manipuladas digitalmente a través de software, pero éste no interviene directamente en su proceso de creación

- **Imágenes Digitales de Síntesis (IDS)**

Las IDS (también llamadas infografías) se sitúan en el entorno de la computación gráfica

Se obtienen a través de software y produciéndolas enteramente a través de algoritmos y sin necesidad de referente real, por ejemplo, realizando dibujos

con el mouse o por medio de programas de render 2D o 3D

Se basan en la introducción de datos matemáticos en la memoria de la computadora que describe y modela. Después se almacena la imagen generada

Las IDS también entran en el terreno de la simulación digital

1	1	1	1	1	1	1	1	1	1
1	0	0	0	1	1	0	0	0	1
1	1	0	1	1	1	1	0	1	1
1	1	0	1	1	1	1	0	1	1
1	1	0	1	1	1	1	0	1	1
1	1	0	0	0	0	0	0	1	1
1	1	0	1	1	1	1	0	1	1
1	1	0	1	1	1	1	0	1	1
1	1	0	1	1	1	1	0	1	1
1	0	0	0	1	1	0	0	0	1
1	1	1	1	1	1	1	1	1	1

Es ventajoso mantener y gestionar los contenidos en formato digital para el mantenimiento, actualización, almacenamiento y acceso

Desplegado de imágenes

Depende de las posibilidades de los sistemas de visualización e impresión

Junto con la velocidad de entrega, los usuarios pueden estar interesados en la calidad de la imagen

Tamaño de pantalla y dimensiones de pixel

A diferencia de los escáneres e impresoras, los monitores actuales ofrecen resolución relativamente baja

Los más típicos son 640x480 o 1,600x1,200

Al aumentar la resolución de la pantalla, se puede ver mayor parte de la imagen

Podemos aumentar la cantidad desplegada de una imagen reduciendo su resolución

Operaciones que se realizan en procesamiento de imágenes

- Edición, retoque o mejora
- Compresión

▼ 19-08-2021

Clases de Imágenes

Imágenes raster (Mapas de bits): Más comunes, formadas por píxeles que se les asigna un color o una intensidad de brillo. Tienen un problema que es el pixelado

Este tipo de imagen se usa para fotografías, ilustraciones o gráficos web. Un editor muy común es el Photoshop

Imágenes vectoriales: No están formadas por píxeles, por lo que son independientes de la resolución y al redimensionar la imagen no se produce pixelado. Están formados por puntos de control que poseen curvas definidas gracias a fórmulas matemáticas que el editor de imágenes se encarga de calcular

- Los vectores son muy utilizados en gráficos de baja complejidad como logotipos, iconos composiciones tipográficas
- Los editores mas usados para crear y modificar gráficos vectoriales son Adobe Illustrator y Corel Draw
- Una imagen vectorial puede ser convertida a mapa de bits en forma muy fácil, pero es muy complicado obtener la calidad de una imagen rasterizada en vectores ya que siempre se pierden detalles

Formatos de imágenes raster

- JPEG: Joint Photographic Experts Group. Se utiliza para trabajos de fotografía porque es el que más colores puede captar. Se puede reducir su tamaño de la imagen sin que influya en su calidad y no admite transparencias. Se recomienda usar este formato para los siguientes escenarios:
 - Imágenes fijas
 - Fotografías
 - Imágenes con muchos colores
 - Imágenes con sombreados, mucha luminosidad y contraste
- GIF: Graphic Intechange Format. Se representan videos y animaciones. Están limitados a 8 bits. Por ello hay solo 256 posibles combinaciones de colores en el formato. Generalmente es más grande que los JPEG. Permite la comprensión rápida y sin esfuerzo de lo que intentan transmitir

▼ 20-08-2021

- PNG: Portable Network Graphics. Formato de gráficos de mapa de bits. Originalmente fue desarrollado como alternativa al GIF. PNG pertenecía a Unisys. Utiliza una paleta de 256 colores. Da la posibilidad de definir también 256 niveles de transparencia mientras que GIF permite que se defina como transparente solo un color de la paleta. También permite mostrar la imagen de forma gradual

Cada fragmento corresponde a 4 partes

- El tamaño: Un entero de 4 bytes
- El tipo de fragmento: Un código de 4 caracteres

- **Los datos del fragmento; la CRC** (comprobación de redundancia cíclica), un código de corrección de 4 bytes que permite comprobar la integridad
- Los fragmentos pueden aparecer en cualquier orden excepto por el hecho de que el fragmento de encabezado debe ir adelante (**fragmento IHDR**) y el de cierre (**fragmento IEND**) detrás
- Los principales fragmentos (denominados **fragmentos críticos**) son IHDR Encabezado de información de mapa de bits, PLTE la paleta, IDAT los datos de la imagen, IEND el cierre de la imagen

- Los otros fragmentos (denominados **fragmentos secundarios**) son los siguientes: bKGD el color de fondo, cHRM las cromaticidades primarias y el punto blanco, gAMA la gama de la imagen, hIST el histograma de la imagen, pHYs
- Las dimensiones del píxel físico, sBIT los bits importantes, tEXt los datos de texto, tIME la hora de la última modificación, tRNS la transparencia y zTXt los datos de texto comprimidos

- TIFF: Tagged Image File Format. Formato de archivos de gráficos de mapa de bits. Compatible con una amplia variedad de plataformas. Pueden almacenar imágenes con un número arbitrario de bits por píxel y pueden emplear diversos algoritmos de compresión. Pueden almacenarse varias imágenes en un único archivo TIFF de varias páginas
- BMP: Bit Map: Formato por excelencia de ráster propio de Windows. Guarda imágenes de 24 bits (16.7 millones de colores), 8 bits (256 tonos de gris) y menos

En esta clase de archivos puede seleccionarse una compresión RLE (Run Length Encoding) sin pérdida de calidad

Formatos de imágenes vectoriales

- AI: Adobe Illustrator. Formato vectorial más usado por defecto del programa de Adobe Illustrator. El formato guarda transparencia del diseño original
- EPS: Encapsulated PostScrip. Formato antiguo de archivos vectoriales. Eps es muy protable y muy compatible con todas las versiones nuevas y antiguas de Adobe Illustrator. No es muy bueno con la cuestión de transparencia
- PDF: Portable Document File. Formato vectorial por excelencia. Propiedad de Adobe. Se trata del formato más universal. Muy útil para intercambiar información
- SVG: Scalable Vector Graphics. Basado en XML. Uso extendido en el diseño web. Puede ser indexado por eso suele usarse para logotipos, botones y otros elementos web

▼ 23-08-2021

Luminancia

La escala de grises es la representación de una imagen en la que cada píxel se dibuja usando un valor numérico individual que representa su luminancia, en una escala que se extiende del blanco al negro

Las escalas de grises también son mapas de bits

Dependiendo del uso, sea tratamiento digital o fotografía, el número de parches varia en torno a los diez para permitir la cantidad de la escala de grises

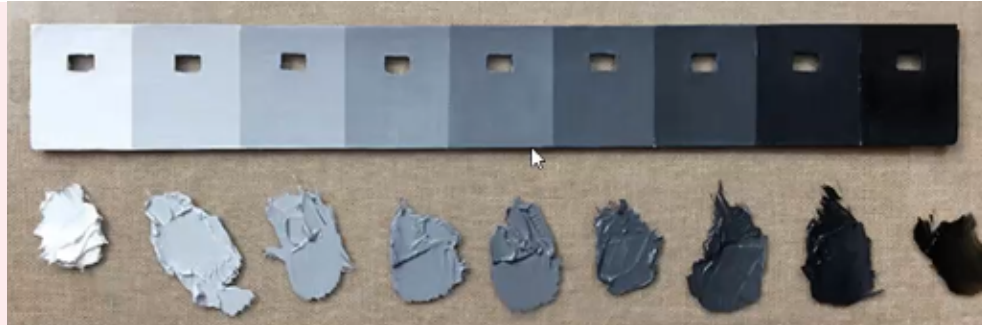
El número de valores que abarcan las escalas es variable

La escala de grises más simple es de 3 variantes: Blanco, gris y negro

Hay de entre 3 y 10 valores

Cuando un tono claro y otro oscuro entran en contacto se producen ilusiones ópticas que modifican el grado de luminosidad con el que ambos son percibidos

Escala de grises de Rodman:



La escala de grises es una grama monocromática

Pueden eliminarse los colores y dejar sólo la luminancia

Pueden promediarse todos los valores del RGB de luminancia de cada píxel

Otro método sería mantener sólo los valores de luminancia del canal rojo, verde o azul

▼ 24-08-2021

Imágenes binivel

También llamada imagen binaria

Es una imagen digital que tiene únicamente dos valores posibles para cada píxel

Representación real del blanco y negro

Normalmente sólo muestran la silueta de los objetos

Se define un umbral y al aplicar una operación de umbral de modo que lo que supere el umbral será blanco y lo que no será negro. De modo que se pueda transformar una escala de grises a blanco y negro

Se pueden calcular propiedades geométricas como el tamaño o la posición de una imagen

Con más de un objeto en el campo de visión, se pueden determinar propiedades topológicas del ensamblaje, como la diferencia entre el número de objetos y el número de huecos

Estas imágenes tienen un alto contraste entre el objeto y el fondo

El patrón obtenido debe ser esencialmente bidimensional

▼ 25-08-2021

Resolución: Cantidad de elementos independientes por unidad de medida lineal

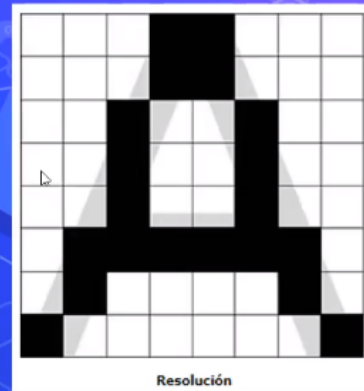
La unidad de medida lineal utilizada para la resolución son los píxeles por pulgada (ppi) que indican los píxeles o celdas por pulgada lineal

Se divide en cuatro tipos:

1. Resolución espacial: Capacidad del sistema para distinguir el objeto más pequeño sobre una imagen
2. Resolución geométrica: Diferencia que hay entre la posición teórica de un pixel y la real
3. Resolución espectral: Número y rango de longitud de onda del espectro electromagnético registrado en cada banda de la imagen
4. Resolución radiométrica: Profundidad o contraste radiométrico. Número de niveles de gris reconocidos por cada banda

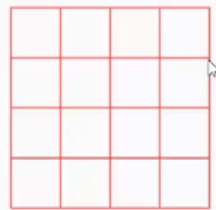
- La resolución es una unidad de **densidad** de imagen, es decir, de concentración de píxeles en línea recta
- Así, una imagen a 72 PPI tendrá 72 celdas distribuidas a lo largo de una línea de 2,54 cm., mientras que una imagen a 300 PPI contendrá 300 celdas a lo largo de una línea de la misma longitud
- Como puede observarse, la longitud de la línea sigue siendo la misma, lo que cambia es el número de celdas o píxeles que se introducen en ese espacio
- Dicho de otro modo, en una imagen a 300 PPI hay mayor concentración de píxeles en una misma unidad de longitud que en una imagen a 72 PPI

- Cuanto mayor sea la resolución mayor será la calidad de la imagen, pues habrá mayor número de celdas para poder definir la forma
- En la imagen se puede ver una simulación de cuadrícula de menor a mayor resolución sobre una letra, y en negro los pixeles que llevarían cada una

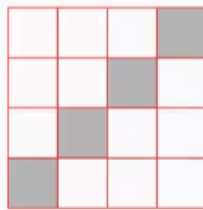


Resolución de imagen, PPI (pixels per inch)

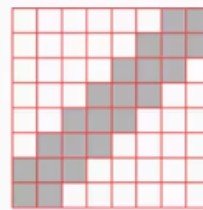
1 PULGADA



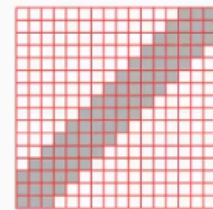
1 cm



4 ppi



8 ppi



16 ppi

- La resolución también se puede expresar en pixels por centímetro (ppcm)
- Para calcular la equivalencia entre ppi y ppcm, se tiene que partir del conocimiento que una pulgada es igual a 2,54 cm, así que dependiendo de la medida que se tenga será dividir entre 2,54 (ppi a ppcm) y multiplicar por 2,54 (ppcm a ppi)
- La resolución hemos dicho que nos indica la calidad de la imagen
- A mayor resolución mayor calidad de la imagen, pero también mayor volumen de información ocupará. La resolución óptima de las imágenes para su posterior impresión es de 300 ppi o 120 ppcm

Tramado digital

- El **tramar** las imágenes es una necesidad por la limitación de la impresión, para poder conseguir el efecto óptico de una imagen de tono continuo
- Para entenderlo de otra forma, hacemos creer ver al ojo un tono continuo donde sólo hay una sucesión de puntos
- Antes se hacía de forma convencional interponiendo una trama entre la imagen original y el material sensible
- Ahora eso se hace mediante el tramado digital

- Con los periféricos de salida, impresora láser y filmadora, se puede “tramar” la imagen
- En el tramado de imágenes se debe tener en cuenta la resolución y lineatura de salida
- Con estos dos parámetros se controla la calidad del trabajo
- Para entender la resolución de salida el concepto es parecido al anteriormente visto
- Es la cantidad de puntos que puede poner en una pulgada

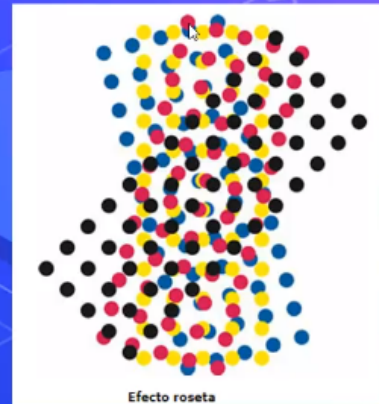
- A mayor resolución de salida más puntos pondrá y mayor definición se obtiene
- Las más habituales son: 1200, 2540, 3600 y 4000 pixeles/pulgada
- La lineatura de salida es el número de puntos por unidad de longitud (pulgadas o centímetros)
- La conversión de líneas por pulgadas a líneas por centímetro y viceversa es igual que lo visto en el apartado anterior
- Las lineaturas más habituales son: 50, 65, 80, 100, 120, 133, 150, 175 y 200 líneas/pulgada

- Existe una pequeña relación entre la resolución y la lineatura a la hora de combinarlas, limitada por los periféricos de salida:

Resolución	Lineatura
1200	50,65,80,100
2540	120,133
3600	150,175
4000	200

- Dentro del tramado, digital y convencional, hay otro concepto que es la frecuencia o ángulo de trama, que se aplica a cada uno de los colores de la cuatricromía: cian (15° ó 105°), magenta (75° ó 165°), amarillo (0° ó 90°) y negro (45° ó 135°)
- Cuando se trabaja con imágenes con dos colores se utilizan las inclinaciones del negro y el magenta, en caso de tres negro, cian y magenta
- El amarillo no se utiliza por que debido al ángulo, siempre dará muaré

- Al aplicar diferentes ángulos a cada color se evita que se produzca un empastamiento de los puntos de la selección de color, y se crea el efecto óptico llamado **roseta**
- Esto se puede ver bien en los carteles del metro y en los displays de los autobuses



- En la figura se ve que los ángulos de la lineatura están formados por círculos
- A esto se le denomina **forma del punto**
- Habitualmente se utilizan tres tipos diferentes
- Redondo, cuadrado y elíptico, habiendo también algunos especiales muy poco usados
- Los redondos dan menor visibilidad y se tocan alrededor del 75% de punto, se suelen utilizar en trabajos a baja lineatura
- Los cuadrados dan mayor sensación de contraste y se tocan alrededor del 50% de punto
- Los elípticos para imágenes más suaves y mayor sensación de continuidad, sus extremos se tocan en porcentajes a partir de 30% de punto



Relación entre resolución de imagen y resolución de impresión

- La resolución de la impresora es distinta de la resolución de imagen, pero sí están relacionadas
- Al imprimir una fotografía de gran calidad en impresoras de inyección de tinta, una resolución de imagen de al menos 220 PPI suele ofrecer resultados aceptables
- Asimismo, existe una clasificación estandarizada de las resoluciones óptimas más frecuentes según el sistema de impresión que se puede usar como guía de referencia:
 - Impresión offset: 300 PPI (imágenes monocromas: 600 PPI)
 - Impresión huecograbado: 300 PPI
 - Impresión flexografía: de 150 a 300 PPI
 - Impresión serigrafía: de 100 a 200 PPI
 - Impresión digital de gran formato: de 72 a 150 PPI
 - Impresión digital de pequeño formato: de 200 a 300 PPI

▼ 26-08-2021

Preliminares de distancia

La distancia más usual en $\mathbb{R}^n$ es la distancia euclídea

La distancia euclídea en $\mathbb{R}^2$ viene dada por la fórmula:

$$d((x_1, y_1), (x_2, y_2)) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

Otras distancias conocidas en $\mathbb{R}^2$ son:

$$d(x, y) = 0 \text{ si } x = y; d(x, y) = 1 \text{ en otro caso (distancia discreta)}$$

$$d((x_1, y_1), (x_2, y_2)) = |x_1 - x_2| + |y_1 - y_2| \text{ (distancia City-block)}$$

$$d((x_1, y_1), (x_2, y_2)) = \max\{|x_1 - x_2|, |y_1 - y_2|\} \text{ (distancia Chessboard)}$$

Conectividad y vecindad

Sea (X, d) un espacio métrico. Se llama bola abierta centrada en un punto x de X de radio $e > 0$ al conjunto:

$$B(x, e) = \{y \in X : d(x, y) < e\}$$

Continuidad

Objetos Homeomorfos

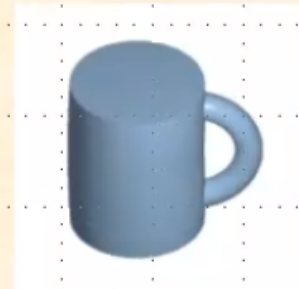
Objetos Homeomorfos

Se dice que dos objetos X e Y son **homeomorfos** (o topológicamente iguales) si existe una función $f: X \rightarrow Y$ tal que f y f^{-1} son biyectivas y continuas.

Ejemplo: un cuadrado y un círculo son homeomorfos. Sin embargo, un círculo y un segmento no lo son.

Dado que es muy difícil comprobar si dos espacios son homeomorfos, existe un concepto más débil que es el de **homotopía**: se dice que dos espacios son **homotópicos** si existe una **deformación continua** que lleve el uno en el otro.

Por ejemplo, aunque un círculo y un segmento no son homeomorfos, sí son homotópicos.



Propiedad topológica

Una propiedad se dice que es **propiedad topológica** si se preserva por homeomorfismo.

Algunas propiedades topológicas:

- Número de componentes conexas
- Número de agujeros
- La compacidad (el hecho de ser cerrado y acotado)
- El número de cavidades (si estamos en 3D)
- ...

Conectividad

Dado un espacio métrico (X, d) , se llama **camino simple** de x a y , a una función continua y biyectiva $f: [0, 1] \rightarrow X$, tal que $f(0)=x$, $f(1)=y$.

Se llama **curva simple** si $x \neq y$.

X se dice que es **conexo** si para cualesquiera par de puntos de X existe un camino que los une.

El teorema de Jordan. Una curva simple en $\mathbb{R}^2$ divide al plano en dos componentes conexas, una acotada y otra no.

Conectividad

- La **conectividad** de pixeles es un concepto utilizado para establecer los limites en objetos y regiones de componentes en una imagen
- Para establecer la conectividad entre dos pixeles, es necesario determinar si son adyacentes en sentido especifico (si tiene 4 vecindades) y su nivel satisface un criterio especificado de similitud (si son iguales)
- Por ejemplo, en una imagen binaria con valores de 0 y 1 dos pixeles pueden tener vecindad de 4, pero solo se consideran conectados si tienen el mismo valor

Vecindad

- Un pixel p en las coordenadas (x, y) tienen 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 pixeles se define como $N_4(p)$

El concepto de vecindad en imágenes digitales

Construiremos una topología consistente en el espacio discreto, además proveeremos definiciones básicas robustas como conectividad y distancia en dicho espacio.

La definición de *topología digital* se basa en la definición de una vecindad en cada píxel.

Llamamos *q-vecindad* ó *q-adyacencia* de un píxel p , $N_q(p)$, al conjunto de píxeles que definimos como vecinos de p .

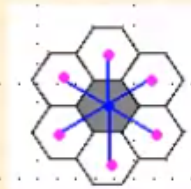
Nos centraremos en el estudio de imágenes digitales binarias. Supondremos que el borde del mallado está compuesto por píxeles blancos. Los objetos de las imágenes estarán formados por píxeles negros y el fondo por píxeles blancos.

Los vecinos de un píxel vienen condicionados por el mallado considerado en la imagen digital.

Veamos los distintos tipos de vecindades para cada tipo de mallado.

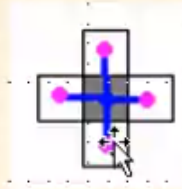
Mallado hexagonal

En este caso particular, definimos la 6-vecindad ó 6-adyacencia de un píxel p como los 6 píxeles cuyas regiones comparten un lado con p .

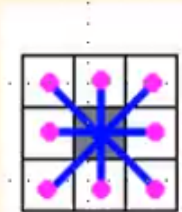


Mallado cuadrangular

La 4-vecindad o 4-adyacencia de un píxel p son los 4 píxeles cuyas regiones comparten un lado con p .

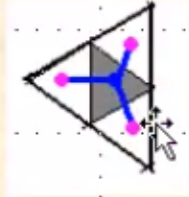


La 8-vecindad u 8-adyacencia de un píxel p , consiste en los 8 píxeles cuyas regiones comparten un lado o un vértice con p .

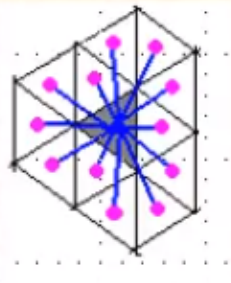


Mallado triangular

La 3-vecindad de un píxel p son los 3 píxeles cuyas regiones comparten un lado con p .



La 12-vecindad de un píxel p , consiste en los píxeles cuyas regiones comparten un lado o un vértice con p .



Caminos digitales

Dada una imagen digital binaria con una relación de vecindad definida (t-adyacencia), un **camino digital** (ó *t-camino*) de un píxel p a otro píxel q se define como una sucesión de píxeles $P_{pq} = \{p_i; i=0, \dots, n\}$ (del mismo color, todos distintos), tal que:

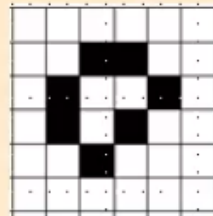
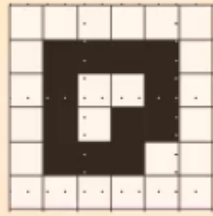
- $p_0 = p, p_n = q$
- Para todo $i=1, \dots, n-1$, p_i tiene exactamente dos vecinos en P_{pq} que son p_{i-1} y p_{i+1}
- p_0 y p_n tienen exactamente un vecino que son p_1 y p_{n-1} , respectivamente.

La longitud de un camino digital con $n+1$ píxeles es n .

Curva digital

Definimos **curva digital** como un conjunto de píxeles tal que al eliminar cualquiera de ellos, se convierte en un camino digital.

Ejemplo:



Curva digital con la 4-adyacencia y la 8 adyacencia, respectivamente.

▼ 27-08-2021

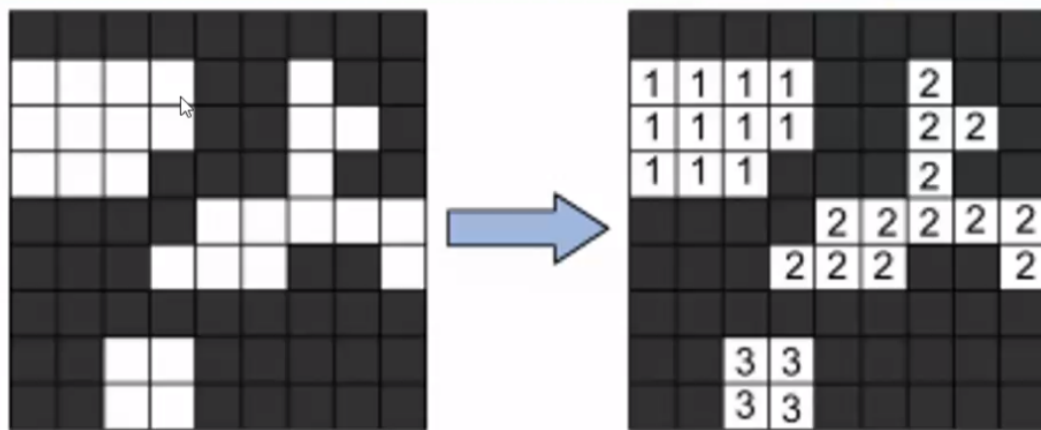
No hubo clase

▼ 30-08-2021

Componentes Conectados

Etiquetado de componentes conectados

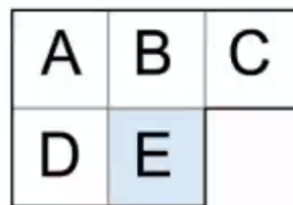
- El etiquetado de **componentes conectados** (o simplemente etiquetado, es una operación que agrupa los píxeles correspondientes al mismo objeto y les asigna una etiqueta, separando así unos objetos de otros
- Este proceso se realiza usualmente cuando la imagen ha sido binarizada previamente
- Como 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



. Imagen original y resultado del etiquetado.

- En este sentido el peor caso que puede plantearse es un objeto con forma de espiral
- Como se aprecia en la figura anterior, 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

- 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:



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

Algoritmo clásico o de dos pasadas

- Una de las primeras publicaciones que describía este algoritmo fue escrita por Rosenfeld y Pflatz
- 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

Algoritmo de múltiples escaneos

- Este algoritmo fue propuesto en 1981 por Haralick, 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, 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, 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

▼ 31-08-2021

Funciones de intensidad

- Como ya se había señalado una imagen es una función bidimensional de la intensidad de luz $f(x,y)$
- Puesto que la luz es una forma de energía, $f(x,y)$ debe estar en el rango,

$$0 < f(x,y) < \infty$$

- La naturaleza básica de $f(x,y)$ puede estar caracterizada por dos componentes:
- La cantidad de luz incidente proveniente de la fuente (componente de iluminación $i(x,y)$)
- La cantidad de luz reflejada por los objetos de la escena (componente de reflejancia $r(x,y)$)

- Las funciones $i(x,y)$ y $r(x,y)$ se combinan en un producto para dar la imagen de función de imagen bidimensional

$$f(x,y) = i(x,y) r(x,y)$$

- Donde,

$$0 < i(x,y) < \infty$$

- Y

$$0 < r(x,y) < 1$$

- A la intensidad de una imagen monocroma $f(x,y)$, en las coordenadas (x,y) se le denomina nivel de gris (I)
- Por lo tanto resulta evidente que I está en el rango:

$$L_{min} < I < L_{max}$$

- Al intervalo se le denomina escala de grises y generalmente adopta los valores $[0,L]$

- Para ser utilizable en procesamiento de imágenes por computadora, una función de imagen $f(x,y)$ debe ser digitalizada tanto espacialmente como en su amplitud
- La digitalización de las coordenadas espaciales se denomina muestreo de la imagen y la digitalización de la amplitud se conoce bajo el nombre de cuantificación del nivel de gris

- Se supone que una imagen continua se describe en forma aproximada por una serie de muestras igualmente espaciadas organizadas en forma de una matriz donde cada elemento de la matriz es una cantidad discreta: $M \times N$

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

- El término de la derecha representa lo que comúnmente se denomina como imagen digital
- A cada elemento de la matriz se le denomina elemento de imagen (pixel)
- Los términos imágenes y píxeles van a emplearse

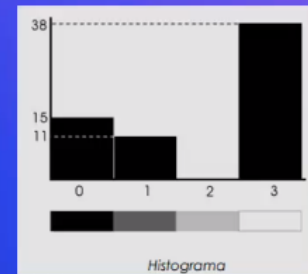
- Es común expresar el muestreo y la cuantificación de una imagen digital en términos matemáticos mas formales
- Si Z y $\mathcal{R}$ representan los conjuntos de números enteros y reales respectivamente, el muestreo puede entenderse como una partición en una cuadrícula del plano siendo las coordenadas del centro de cada elemento de la cuadrícula un par de elementos del producto cartesiano $Z \times Z$ que es el conjunto de todos los pares ordenados (a,b)
- Siendo a y b enteros pertenecientes a Z

- Por lo $f(x,y)$ representa una imagen digital si (x,y) son enteros de $Z \times Z$ y f es una función que asigna un nivel de gris
- En general el ojo humano puede distinguir alrededor de 30 niveles :

▼ 01-09-2021

Funciones de intensidad

- La gráfica que aparece en la siguiente figura es el histograma correspondiente a esta imagen
- Los números que aparecen en el eje horizontal representan los niveles de gris que pueden aparecer en la imagen: a la izquierda está el valor más oscuro (negro) y en el extremo derecho el más claro (blanco)
- El resto de niveles se distribuyen uniformemente
- Se ha puesto una escala con los tonos de gris correspondientes para facilitar la comprensión



7

- ¿Cuánto deberán sumar las alturas de todas las barras?
- Efectivamente, 64, que es el número total de píxeles que tiene la imagen
- Como se ve, con sólo mirar el histograma se pueden deducir algunas cosas interesantes sobre la imagen, demostrando su utilidad:
 - La mayor parte de los píxeles son blancos, así que probablemente se aprecie un fondo blanco uniforme
 - Hay un número significativo de píxeles totalmente blancos y totalmente negros, por lo que presentará un aspecto bien contrastado

Histogramas

- Suponer una imagen en niveles de grises, siendo el rango de 256 colores (de 0 a 255)
- El **histograma** de la imagen consiste en una gráfica donde se muestra el número de píxeles de cada nivel de gris que aparecen en la imagen
- En el siguiente ejemplo podemos ver tres imágenes con sus correspondientes histogramas

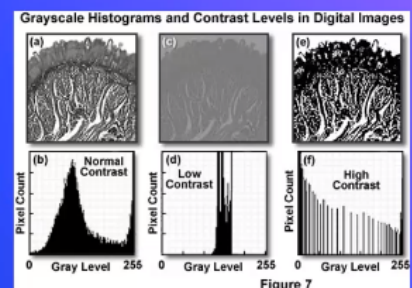
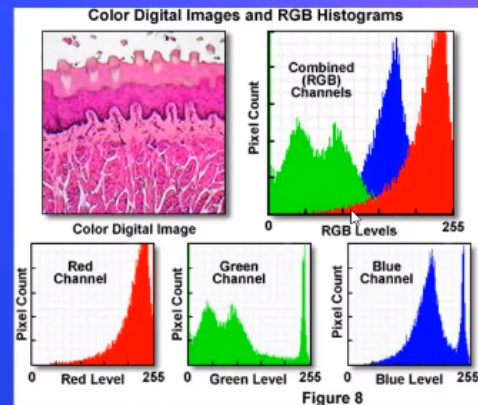


Figure 7

Histogramas

El histograma de una imagen a color RGB consiste en tres gráficas siendo cada una el histograma de cada color primario:



¿Cómo se obtiene el histograma en la práctica?

- La mayoría de los programas de edición de imágenes (o retoque fotográfico) ofrecen la opción de mostrar un histograma de la imagen
- Si se trata de un histograma correspondiente a una imagen con 256 niveles de brillo, su trazado es mucho más continuo (hay 256 barras verticales)
- Como ayuda, se divide el eje horizontal en cinco zonas, que podrían denominarse, de izquierda a derecha: negros, grises oscuros, grises medios, grises claros y blancos.

¿Hay un histograma ideal?

- Cada imagen tiene su propio histograma
- Pero como regla general se considera que una imagen tiene un buen contraste si su histograma se extiende ocupando casi todo el rango de tonos

▼ 02-09-2021

- El **filtrado** es una técnica para modificar o mejorar una imagen
- Por ejemplo, se puede filtrar una imagen para enfatizar ciertas características o eliminar otras características
- Las operaciones de procesamiento de imágenes implementadas con filtrado incluyen **suavizado, nitidez y mejora de bordes**
- El filtrado es una *operación de vecindad*, en la que el valor de cualquier píxel dado en la imagen de salida se determina aplicando algún algoritmo a los valores de los píxeles en la vecindad del píxel de entrada correspondiente

Convolución

Convolución

- La convolución es una operación de vecindad en la que cada píxel de salida es la suma ponderada de los píxeles de entrada vecinos
- La matriz de pesos se llama núcleo de convolución , también conocido como filtro
- Un kernel de convolución es un kernel de correlación que se ha girado 180 grados

- Por ejemplo, suponga que la imagen es

$$A = \begin{bmatrix} 17 & 24 & 1 & 8 & 15 \\ 23 & 5 & 7 & 14 & 16 \\ 4 & 6 & 13 & 20 & 22 \\ 10 & 12 & 19 & 21 & 3 \\ 11 & 18 & 25 & 2 & 9 \end{bmatrix}$$

- y el kernel de correlación es

$$h = \begin{bmatrix} 8 & 1 & 6 \\ 3 & 5 & 7 \\ 4 & 9 & 2 \end{bmatrix}$$

Utilizaría los siguientes pasos para calcular el píxel de salida en la posición (2,4):

- Gire el núcleo de correlación 180 grados sobre su elemento central para crear un núcleo de convolución
- Deslice el elemento central del núcleo de convolución de modo que quede encima del elemento (2,4) de A
- Multiplique cada peso en el núcleo de convolución girado por el píxel de A debajo.
- Sume los productos individuales del paso 3
- Por lo tanto, el píxel de salida (2,4) es
- Se muestra en la siguiente figura

Correlación

- Los filtros espaciales pueden clasificarse basándose en su linealidad en **filtros lineales** y en **filtros no lineales**
- A su vez los filtros lineales pueden ser clasificados según las frecuencias que dejen pasar: los filtros **paso bajo** atenúan o eliminan las componentes de alta frecuencia a la vez que dejan inalteradas las bajas frecuencias; los filtros **paso altas** atenúan o eliminan las componentes de baja frecuencia con lo que agudizan las componentes de alta frecuencia; los filtros **paso banda** eliminan regiones elegidas de frecuencias intermedias

A continuación se describe el uso de los diferentes filtros:

- Filtros **paso bajas**: son utilizados en la reducción de ruido; suavizan y aplanan un poco las imágenes y como consecuencia se reduce o se pierde la nitidez. En inglés son conocidos como *Smoothing Spatial Filters*

- Filtros **paso banda**: son utilizados para detectar patrones de ruido. Ya que un filtro paso banda generalmente elimina demasiado contenido de una imagen casi no son usados, sin embargo, los filtros paso banda son útiles para aislar los efectos de ciertas bandas de frecuencias seleccionadas sobre una imagen. De esta manera, estos filtros ayudan a simplificar el análisis de ruido, razonablemente independiente del contenido de la imagen

Enfoque	Desenfoque	Realce de bordes	Repujado
$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} -2 & -1 & 0 \\ -1 & 1 & 1 \\ 0 & 1 & 2 \end{bmatrix}$
Detección de bordes	Filtro de tipo Sobel	Filtro de tipo Sharpen	
$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & -2 & 1 \\ -2 & 5 & -2 \\ 1 & -2 & 1 \end{bmatrix}$	
Filtro Norte	Filtro Este	Filtro de tipo Gauss	
$\begin{bmatrix} 1 & 1 & 1 \\ 1 & -2 & 1 \\ -1 & -1 & -1 \end{bmatrix}$	$\begin{bmatrix} -1 & 1 & 1 \\ -1 & -2 & 1 \\ -1 & 1 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 2 & 3 & 1 & 1 \\ 2 & 7 & 11 & 7 & 2 \\ 3 & 11 & 17 & 11 & 3 \\ 2 & 7 & 11 & 7 & 2 \\ 1 & 2 & 3 & 2 & 1 \end{bmatrix}$	

Tabla 2 – Máscaras más usadas.

- Suavizado

1	1	1	1	1
1	4	4	4	1
1	4	12	4	1
1	4	4	4	1
1	1	1	1	1



▼ 03-09-2021

No hubo clase

▼ 06-09-2021

Continuación de filtros

Recordando: los filtros son operaciones sobre los pixeles considerando una ponderación en la vecindad en cada uno de ellos

Filtros especiales

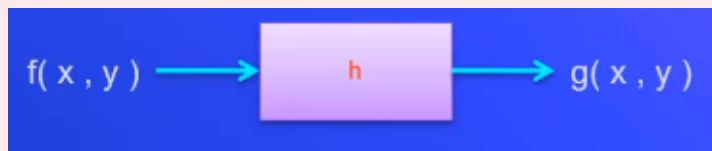
Filtros en el dominio del espacio:

⬡ Filtros

- Filtros espaciales
 - Filtros lineales
 - Filtros pasa bajos
 - Filtro pasa altos
 - Filtro pasa bandas
 - Filtros no lineales (de orden estadístico)
 - Filtro max
 - Filtro min
 - Filtro mediana

Filtros no lineales o de orden estadístico

Es un tipo de operación que altera el valor de un píxel en función de los valores de los píxeles que le rodean



También se le denomina procesamiento basado en la vecindad u operación de vecindad

Filtrar una imagen consiste en aplicar una transformación de forma que se acentúen o disminuyan ciertos aspectos

- Los filtros de orden estadístico son filtros espaciales no lineales cuya respuesta está basada en el ordenamiento de los píxeles contenidos en la imagen que encierra el filtro, y que reemplazan el valor del píxel central por el valor determinado por dicho ordenamiento

Filtro mediana

- El ejemplo más conocido es el filtro **mediana**, que reemplaza el píxel central por la mediana de los píxeles ordenados contenidos en la vecindad del filtro (subventana)
- Los filtros mediana eliminan cierto tipo de ruido de manera muy eficiente, el llamado ruido impulso o **sal y pimienta**

○ Ruido sal y pimienta

Ruido provocado durante la transmisión de la imagen

- **Ruido impulsivo o ruido de sal y pimienta** (*salt-pepper noise*).
 - También aparece con frecuencia durante la transmisión de la imagen
 - Hay **puntos** de la imagen con niveles de intensidad muy diferentes a los de su entorno.
 - Suele aparecer en forma de impulsos **blancos** o **negros**.

$$g(x, y) = \begin{cases} \eta(x, y) & \text{con probabilidad } p \\ f(x, y) & \text{con probabilidad } 1 - p \end{cases}$$

- Los impulsos pueden tener valores prefijados (v.g.: 0 ó 255) o pueden tener una distribución de probabilidad de "cola larga"

- La **mediana**, ξ , de un conjunto ordenado de valores es aquel valor para la cual la mitad del conjunto está por debajo de ella y la otra mitad por arriba
- Por ejemplo, en una vecindad de 3×3 la mediana del conjunto ordenado sería el 5to elemento del conjunto
- Si los valores son: {10, 20, 20, 20, 15, 20, 20, 25, 100} y el conjunto ordenado es: {10, 15, 20, 20, 20, 20, 20, 25, 100}, la mediana es: $\xi=20$

- La función principal del filtro mediana es forzar a puntos que son diferentes de los niveles de gris del vecindario a ser más parecidos a él
- Si el tamaño de la sub-ventana del filtro es grande entonces la imagen se puede convertir en borrosa (efecto de pintado)

Otro ejemplo es el siguiente:

Ejemplo (Filtro de la mediana)

Considérese la sub-ventana de 3×3

100	89	92
110	25	80
95	78	93

El ordenamiento de los valores es

25, 78, 80, 89, 92, 93, 95, 100, 110

El nuevo valor del pixel central es 92.

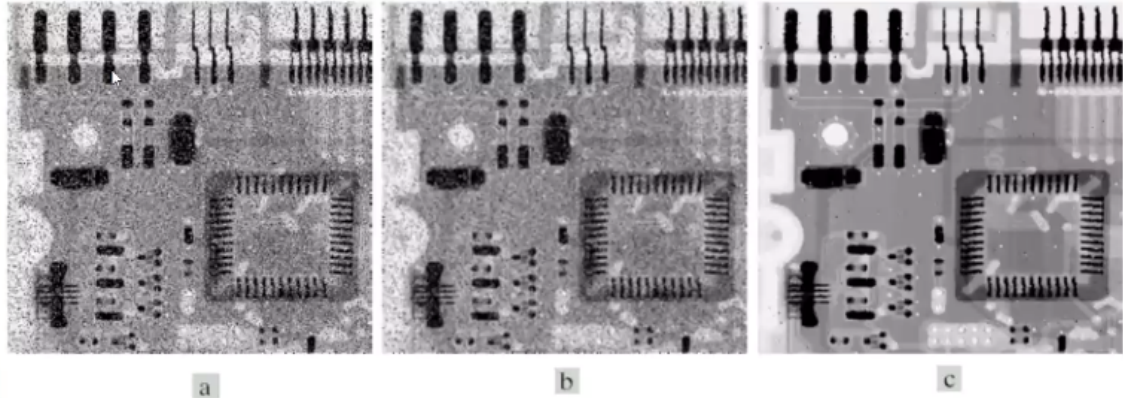


Imagen de rayos X de un
circuito integrado corrupto
con mucho ruido del tipo
"sal y pimienta"

Filtro promedio de 3 x 3

Filtro mediana de 3 x 3,
se nota la superioridad
respecto a b

- El filtro mediana es el más utilizado de los filtros de orden estadístico no lineales pero no es el único
- Existe el filtro **max** que es útil para encontrar los puntos más brillantes en una imagen: donde para un filtro de 3 x 3, $R = \max\{z_k \mid k=1, 2, \dots, 9\}$
- Este filtro elige el valor del pixel que ocupa la posición más alta o máxima en el ordenamiento de cada sub-ventana

Ejemplo (Filtro del máximo)

Considérese la sub-ventana de 3×3

$$\begin{bmatrix} 100 & 89 & 92 \\ 110 & 25 & 80 \\ 95 & 78 & 93 \end{bmatrix}$$

El ordenamiento de los valores es

25, 78, 80, 89, 92, 93, 95, 100, 110

El nuevo valor del pixel central es 110.

- Este filtro hace que la imagen sea más clara

Filtro del máximo



Ruido de pimienta
Probabilidad = 0.04



Máscara 3×3

Ejemplo (Filtro de la mínimo)

Considérese la sub-ventana de 3×3

$$\begin{bmatrix} 100 & 89 & 92 \\ 110 & 226 & 80 \\ 95 & 78 & 93 \end{bmatrix}$$

El ordenamiento de los valores es

78, 80, 89, 92, 93, 95, 100, 110, 226

El nuevo valor del pixel central es 78.

- Este filtro hace que la imagen sea más oscura



Instalar MATLAB

Examen de 2 horas

▼ 07-09-2021

MATLAB tiene soporte de herramientas para procesar imágenes

MATLAB admite los siguientes formatos de archivo de gráficos:

- BMP (mapa de bits de Microsoft Windows)
- HDF (formato de datos jerárquico)
- JPEG (Grupo conjunto de expertos en fotografía)
- PCX (pincel)
- PNG (gráficos de red portátiles)
- TIFF (formato de archivo de imagen con etiquetas)
- XWD (X Window Dump)

○ Para obtener información sobre las profundidades de bits y los tipos de imágenes compatibles con estos formatos, consulte *imread* e *imwrite*.

○ MATLAB admite tres clases numéricas diferentes para la visualización de imágenes: punto flotante de doble precisión (doble), entero sin signo de 16 bits (uint16) y entero sin signo de 8 bits (uint8).

○ Los comandos de visualización de imágenes interpretan los valores de los datos de manera diferente según la clase numérica en la que se almacenan los datos.

○ Este capítulo analiza los diferentes tipos de datos e imágenes que puede utilizar, e incluye detalles sobre cómo leer, escribir, trabajar y mostrar imágenes gráficas; cómo modificar las propiedades de visualización y la relación de aspecto de una imagen durante la visualización; cómo imprimir una imagen; y cómo convertir el tipo de datos o formato gráfico de una imagen.

Esta tabla enumera las funciones discutidas

Function	Purpose	Function Group
axis	Plot axis scaling and appearance	Display
image	Display image (create image object)	Display
imagesc	Scale data and display as image	Display
imread	Read image from graphics file	File I/O
imwrite	Write image to graphics file	File I/O
imfinfo	Get image information from graphics file	Utility
ind2rgb	Convert indexed image to RGB image	Utility

Imágenes en MATLAB

La estructura de datos básica en MATLAB es la matriz, un conjunto ordenado de elementos reales o complejos.

Este objeto se adapta naturalmente a la representación de imágenes, conjuntos ordenados de valores reales de color o datos de intensidad. (MATLAB no admite imágenes de valores complejos).

MATLAB almacena la mayoría de las imágenes como matrices bidimensionales (es decir, matrices), en las que cada elemento de la matriz corresponde a un solo píxel en la imagen mostrada

Por ejemplo, puede seleccionar un solo píxel de una matriz de imagen utilizando subíndices de matriz normal

`I(2,15)`

Este comando devuelve el valor del píxel en la fila 2, columna 15 de la imagen I.

Soporte de profundidad de bits

- MATLAB admite la lectura de las profundidades de bits más utilizadas (bits por píxel) de cualquiera de los formatos de archivo de gráficos admitidos
Cuando los datos están en la memoria, se pueden almacenar como uint8, uint16 o double
Para obtener detalles sobre qué profundidades de bits son apropiadas para cada formato admitido, consulte `imread` e `imwrite`

- Los datos de estas matrices se almacenan como números de punto flotante de doble precisión (64 bits)
Todas las funciones y capacidades de MATLAB funcionan con estos arreglos
- Sin embargo, para imágenes almacenadas en uno de los formatos de archivo de gráficos compatibles con MATLAB, esta representación de datos no siempre es ideal
El número de píxeles en una imagen de este tipo puede ser muy grande; por ejemplo, una imagen de 1000 por 1000 tiene un millón de píxeles

Tipos de imágenes

En MATLAB, una imagen consta de una matriz de datos y posiblemente una matriz de mapa de colores

En MATLAB se utilizan tres tipos de imágenes básicas, cada uno de los cuales difiere en la forma en que se interpretan los elementos de la matriz de datos:

- Imágenes indexadas
- Imágenes de intensidad (o escala de grises)
- Imágenes RGB (o color verdadero)

Esta sección analiza cómo MATLAB representa cada uno de estos tipos de imágenes

Imágenes indexadas

Una imagen indexada consta de una matriz de datos, `X`, y una matriz de mapa de colores, `map`. `map` es una matriz de `m-por-3` de clase `double` que contiene valores de punto flotante en el rango `[0, 1]`.

Cada fila del mapa especifica los componentes rojo, verde y azul de un solo color.

Una imagen indexada utiliza un "mapeo directo" de los valores de los píxeles a los valores del mapa de colores.

El color de cada píxel de la imagen se determina utilizando el correspondiente valor de `X` como índice en el mapa

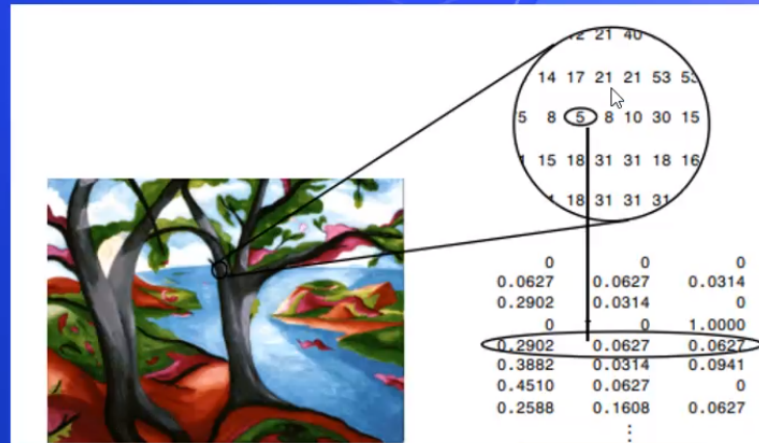
El valor 1 apunta a la primera fila del mapa, el valor 2 apunta a la segunda fila, y así sucesivamente.

Puede mostrar una imagen indexada con las declaraciones `image(X); colormap(map)`

Un mapa de colores a menudo se almacena con una imagen indexada y se carga automáticamente con la imagen cuando usa la función de lectura.

Sin embargo, no está limitado a utilizar el mapa de colores predeterminado; puede utilizar cualquier mapa de colores que elija. La descripción de la propiedad `CDataMapping` describe cómo modificar el tipo de mapeo utilizado

La siguiente figura ilustra la estructura de una imagen indexada.
 Los píxeles de la imagen están representados por números enteros, que son punteros (índices) a los valores de color almacenados en el mapa de colores.



▼ 08-09-2021

Continuación de la clase

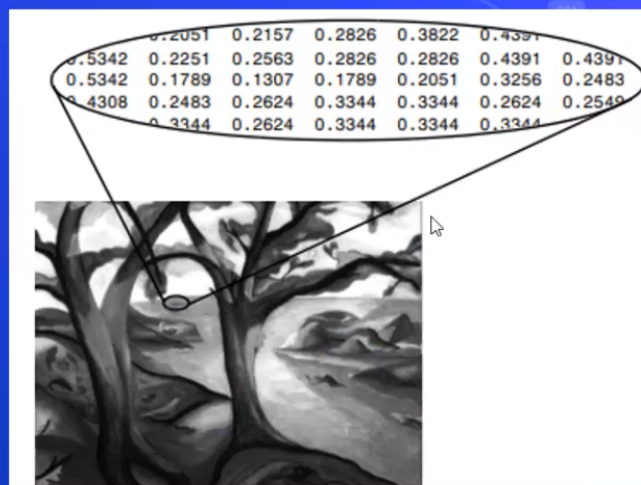
- La relación entre los valores en la matriz de la imagen y el mapa de colores depende de la clase de la matriz de la imagen
- Si la matriz de la imagen es de clase double, el valor 1 apunta a la primera fila del mapa de colores, el valor 2 apunta a la segunda fila, y así sucesivamente
- Si la matriz de la imagen es de clase uint8 o uint16, hay un desplazamiento: el valor 0 apunta a la primera fila en el mapa de colores, el valor 1 apunta a la segunda fila, y así sucesivamente

- El desplazamiento también se utiliza en formatos de archivos gráficos para maximizar la cantidad de colores que se pueden admitir
- En la imagen anterior, la matriz de la imagen es de clase double
- Como no hay desplazamiento, el valor 5 apunta a la quinta fila del mapa de colores

Imágenes de intensidad

- Una imagen de intensidad es una matriz de datos, I , cuyos valores representan intensidades dentro de algún rango
- MATLAB almacena una imagen de intensidad como una única matriz, y cada elemento de la matriz corresponde a un píxel de imagen
- La matriz puede ser de clase double, uint8 o uint16.
- Si bien las imágenes de intensidad rara vez se guardan con un mapa de colores, MATLAB usa un mapa de colores para mostrarlas
- En esencia, MATLAB maneja las imágenes de intensidad como imágenes indexadas

Esta figura muestra una imagen de intensidad de clase doble.



- Para mostrar una imagen de intensidad, utilice la función `imagesc` (“escala de imagen”), que le permite establecer el rango de valores de intensidad.
- `imagesc` escala los datos de la imagen para utilizar el mapa de colores completo.
- Utilice la forma de imágenes de dos entradas `imagesc` para mostrar una imagen de intensidad.
- Por ejemplo,

```
imagesc(I,[0 1]); colormap(gray);
```
- El segundo argumento de entrada a `imagesc` especifica el rango de intensidad deseado

- La función `imagesc` muestra `I` asignando el primer valor en el rango (generalmente 0) a la primera entrada del mapa de colores, y el segundo valor (generalmente 1) a la última entrada del mapa de colores. Los valores intermedios se distribuyen linealmente en el resto de colores del mapa de colores
- Aunque es convencional mostrar imágenes de intensidad usando un mapa de colores en escala de grises, es posible usar otros mapas de colores

Imágenes RGB (Truecolor)

- Una imagen RGB, a veces denominada imagen de “color verdadero”, se almacena en MATLAB como una matriz de datos m-por-n-por-3 que define los componentes de color rojo, verde y azul para cada píxel individual
 - Las imágenes RGB no utilizan paleta
 - El color de cada píxel está determinado por la combinación de las intensidades de rojo, verde y azul almacenadas en cada plano de color en la ubicación del píxel
 - Los formatos de archivo de gráficos almacenan imágenes RGB como imágenes de 24 bits, donde los componentes rojo, verde y azul son de 8 bits cada uno
-
- Esto produce un potencial de 16 millones de colores
 - La precisión con la que se puede replicar una imagen de la vida real ha dado lugar al sobrenombre de “*imagen en color verdadero*”
 - Una matriz RGB MATLAB puede ser de clase double, uint8 o uint16
 - En una matriz RGB de clase double, cada componente de color es un valor entre 0 y 1
 - Un píxel cuyos componentes de color son (0,0,0) se muestra como negro y un píxel cuyos componentes de color son (1,1,1) se muestra en blanco
 - Los tres componentes de color para cada píxel se almacenan a lo largo de la tercera dimensión de la matriz de datos

Trabajar con imágenes de 8 y 16 bits

- MATLAB suele trabajar con números de punto flotante de doble precisión (64 bits).
- Sin embargo, para reducir los requisitos de memoria para trabajar con imágenes, MATLAB proporciona soporte para almacenar imágenes como enteros sin signo de 8 o 16 bits mediante el uso de las clases numéricas uint8 o uint16, respectivamente
- Una imagen cuya matriz de datos tiene la clase uint8 se denomina imagen de 8 bits; una imagen cuya matriz de datos tiene la clase uint16 se denomina imagen de 16 bits.

- La función de imagen puede mostrar imágenes de 8 o 16 bits directamente sin convertirlas a doble precisión
- Sin embargo, la imagen interpreta los valores de la matriz de forma ligeramente diferente cuando la matriz de la imagen es uint8 o uint16
- La interpretación específica depende del tipo de imagen

▼ 09-09-2021

No hubo clase

▼ 10-09-2021

No hubo clase

▼ 13-09-2021

Tema 1.1 hasta Tema 2.3 Filtrado Espacial

Trabajar con imágenes de 8 y 16 bits

- MATLAB suele trabajar con números de punto flotante de doble precisión (64 bits).
- Sin embargo, para reducir los requisitos de memoria para trabajar con imágenes, MATLAB proporciona soporte para almacenar imágenes como enteros sin signo de 8 o 16 bits mediante el uso de las clases numéricas uint8 o uint16, respectivamente
- Una imagen cuya matriz de datos tiene la clase uint8 se denomina imagen de 8 bits; una imagen cuya matriz de datos tiene la clase uint16 se denomina imagen de 16 bits.

Imágenes indexadas de 8 y 16 bits

Si la clase de X es uint8 o uint16, sus valores se compensan en 1 antes de usarse como índices de mapa de colores.

- El valor 0 apunta a la primera fila del mapa de colores, el valor 1 apunta a la segunda fila, y así sucesivamente.
- El comando de imagen proporciona automáticamente el desplazamiento adecuado, por lo que el método de visualización es el mismo si X es doble, uint8 o uint16

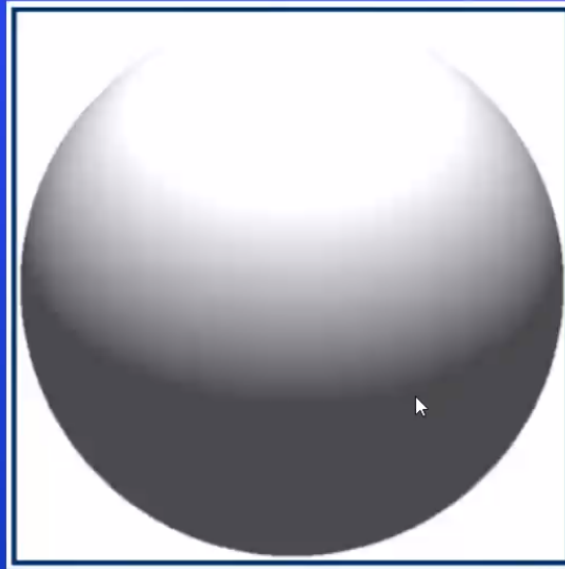
```
image(X); colormap(map);
```

- El desplazamiento del índice del mapa de colores para los datos uint8 y uint16 está diseñado para admitir formatos de archivo de gráficos estándar, que normalmente almacenan datos de imagen en forma indexada con un mapa de colores de 256 entradas.
- El desplazamiento le permite manipular y mostrar imágenes de esta forma en MATLAB utilizando las matrices uint8 y uint16 más eficientes en memoria.
- Debido al desplazamiento, debe agregar 1 para convertir una imagen indexada uint8 o uint16 a doble

Es posible hacer operaciones matemáticas con estos formatos

▼ 14-09-2021

Representación de una imagen original en escala de grises



● Pixel → INTENSIDAD PROMEDIO → número

negro

0



blanco

$(2^n - 1)$

● Pixel → INTENSIDAD PROMEDIO → número

negro
0



blanco
($2^n - 1$)

0

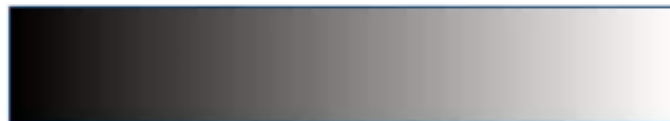


$2^2 - 1 = 3$

$n = 2$
4 niveles

● Pixel → INTENSIDAD PROMEDIO → número

negro
0



blanco
($2^n - 1$)

0



$2^3 - 1 = 7$

$n = 3$
8 niveles

- Pixel → INTENSIDAD PROMEDIO → número

negro

0



blanco

$(2^n - 1)$

0



$2^4 - 1 = 15$

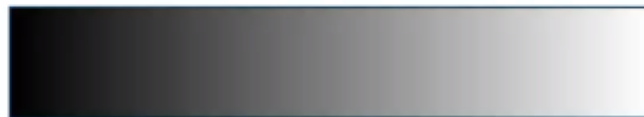
$n = 4$

16 niveles

- Pixel → INTENSIDAD PROMEDIO → número

negro

0



blanco

$(2^n - 1)$

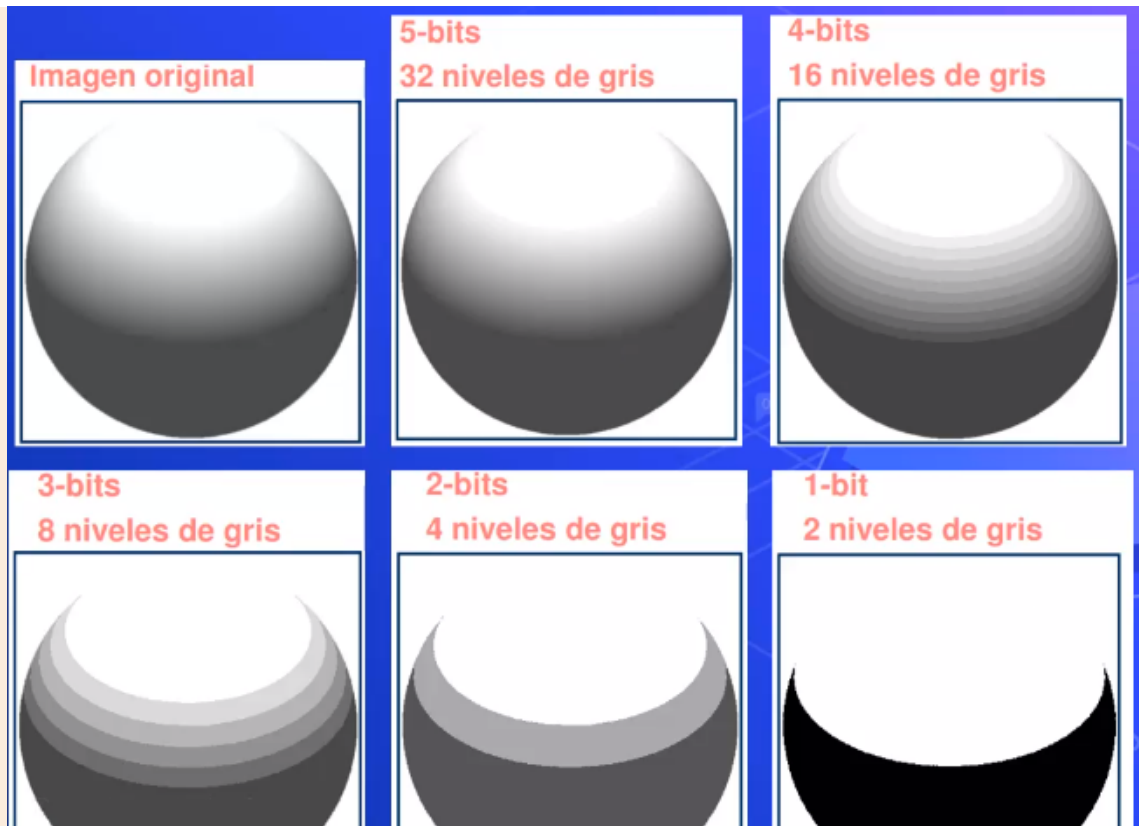
0



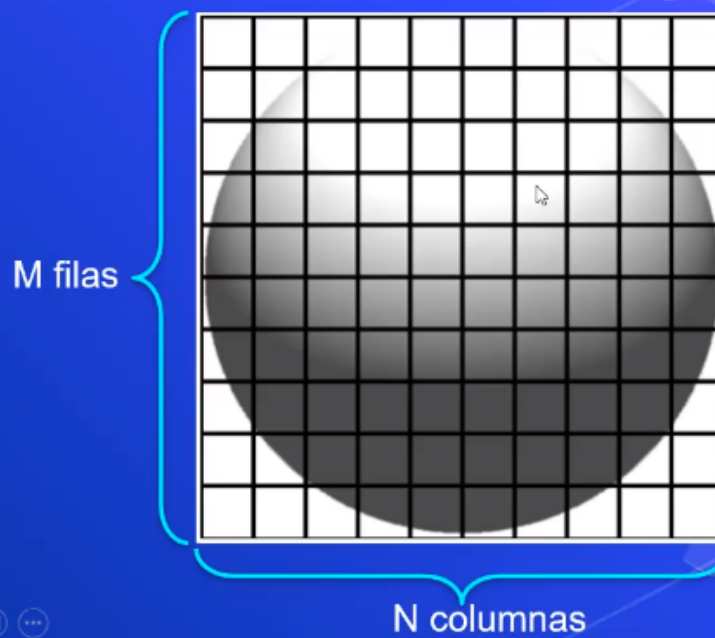
$2^5 - 1 = 31$

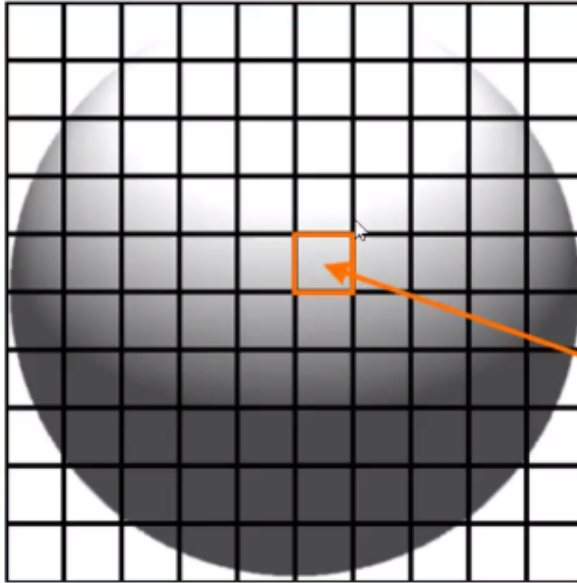
$n = 5$

32 niveles



Representación en formato de matriz (M filas x N columnas)





Cada elemento de la grilla es un **PIXEL**

255	255	254	255	255	255	255	255	254	255	255
255	244	254	255	255	255	255	255	254	244	255
236	223	254	255	255	255	255	255	254	225	234
184	200	232	247	252	252	247	233	202	182	
131	163	190	206	214	214	207	191	164	128	
103	116	142	157	165	165	158	143	117	99	
134	78	91	104	111	111	105	92	78	129	
214	76	75	75	76	76	75	75	76	210	
255	177	76	75	75	75	75	76	172	255	
255	255	214	133	95	95	132	211	255	255	

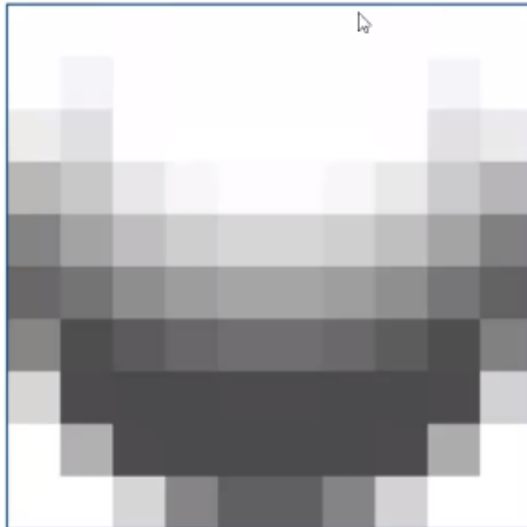
Grillado

$M \times N = 10 \times 10$

8-bits

0 = NEGRO

255 = BLANCO



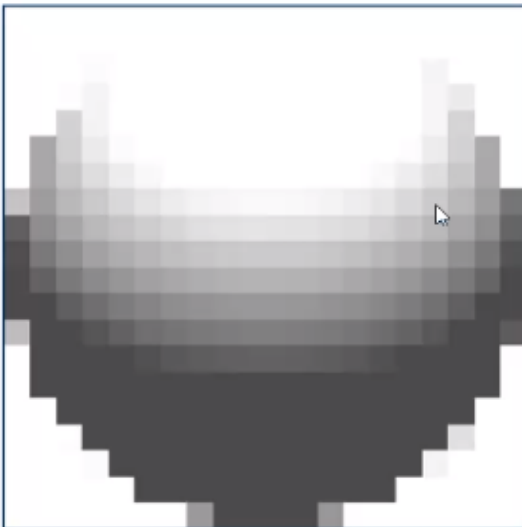
Grillado

$M \times N = 10 \times 10$

8-bits

0 = NEGRO

255 = BLANCO



Grillado

$M \times N = 20 \times 20$

8-bits

0 = NEGRO

255 = BLANCO

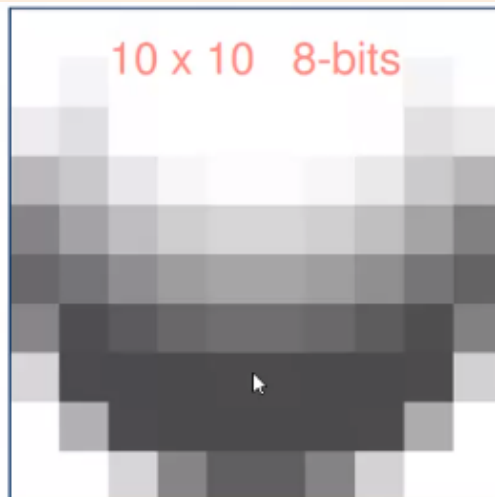
Resolución:

- Es la cantidad de píxeles que definen la imagen.



Ejemplos:

640x480, 800x600, 1024x768, etc.



1,2 kB



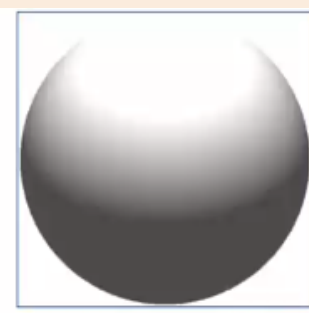
1,4 kB



2,6 kB



124,0 kB



350 x 350 8-bits
 $350 \times 350 = 122.500$ pixels

▼ Segundo Parcial

▼ 20-09-2021

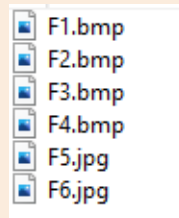
Con MATLAB asegurarse que tenga la herramienta de Procesamiento de Imágenes

Opción:

<https://drive.google.com/drive/folders/1nuuT9PRUN92atEh7LGSLzcWA19xcBYGE?usp=sharing>

▼ 21-09-2021

Tenemos la siguiente carpeta de MATLAB



Y usamos estos comandos

```
%-- 21/09/2021 11:33 a. m. --%
%-- 21/09/2021 12:12 p. m. --%

f1 = imread(F1.bmp)
f1 = imread("F1.bmp")
f1 = imread('F1.bmp')
imshow(f1)

f2 = imread('F2.bmp')
imshow(f2)

f3 = imread('F3.bmp')
imshow(f3)

f4 = imread('F4.bmp')
imshow(f4)

f5 = imread('F5.jpg')
imshow(f5)

f6 = imread('F6.jpg')
imshow(f6)
```

▼ 22-09-2021

Ahora se va a trabajar con un png

Desplegado de imágenes binarias

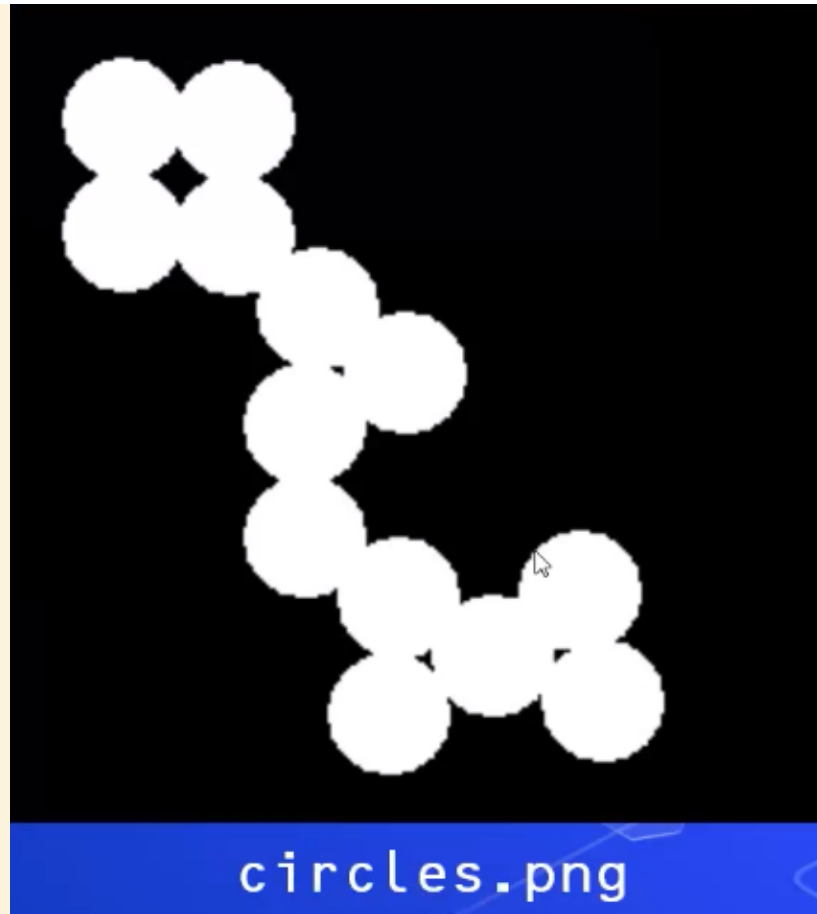
- En MATLAB, una imagen binaria es de clase lógica, las imágenes binarias contienen solo 0 y 1
- Los píxeles con el valor 0 se muestran en negro; los píxeles con el valor 1 se muestran en blanco

Nota Para que la caja de herramientas interprete la imagen como binaria, debe ser de clase lógica

- Las imágenes en escala de grises que contienen solo 0 y 1 no son imágenes binarias

- Para mostrar una imagen binaria, llamar a la función `imshow` o abrir la aplicación Visor de imágenes.
- Por ejemplo, este código lee una imagen binaria en el espacio de trabajo de MATLAB y luego muestra la imagen
- Se va a utilizar el nombre de variable `BW` para representar una imagen binaria en el espacio de trabajo

```
BW = imread('circles.png');  
imshow(BW)
```



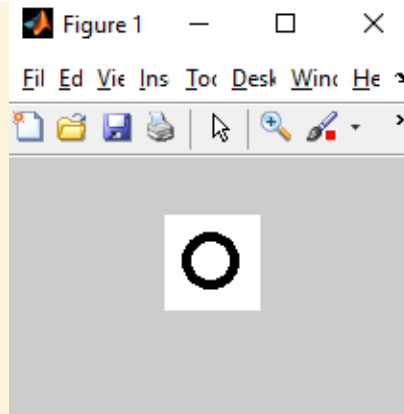
Usamos este comando:

```
>> circle = imread('circle.bmp')
```

Y luego la mostramos:

```
imshow(circle)
```

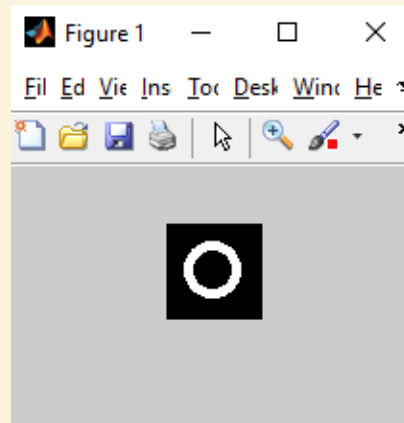
Con este resultado



Al usar este comando

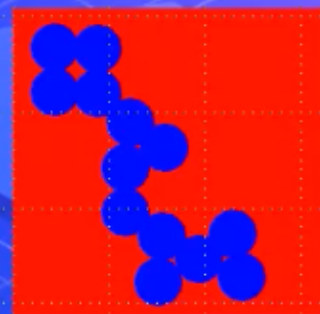
```
imshow(~circle)
```

Obtenemos la imagen negativa



- También puede mostrar una imagen binaria utilizando la sintaxis del mapa de color de imagen indexada
- Por ejemplo, el siguiente comando especifica un mapa de color de dos filas que muestra 0 como rojo y 1 como azul

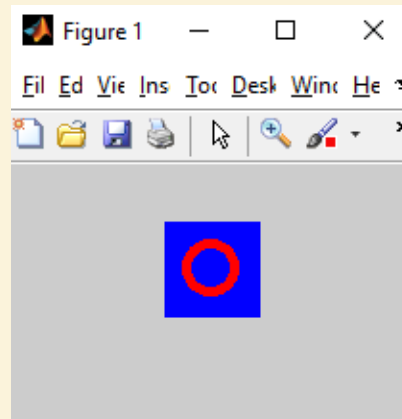
```
imshow(BW,[1 0 0; 0 1 0])
```



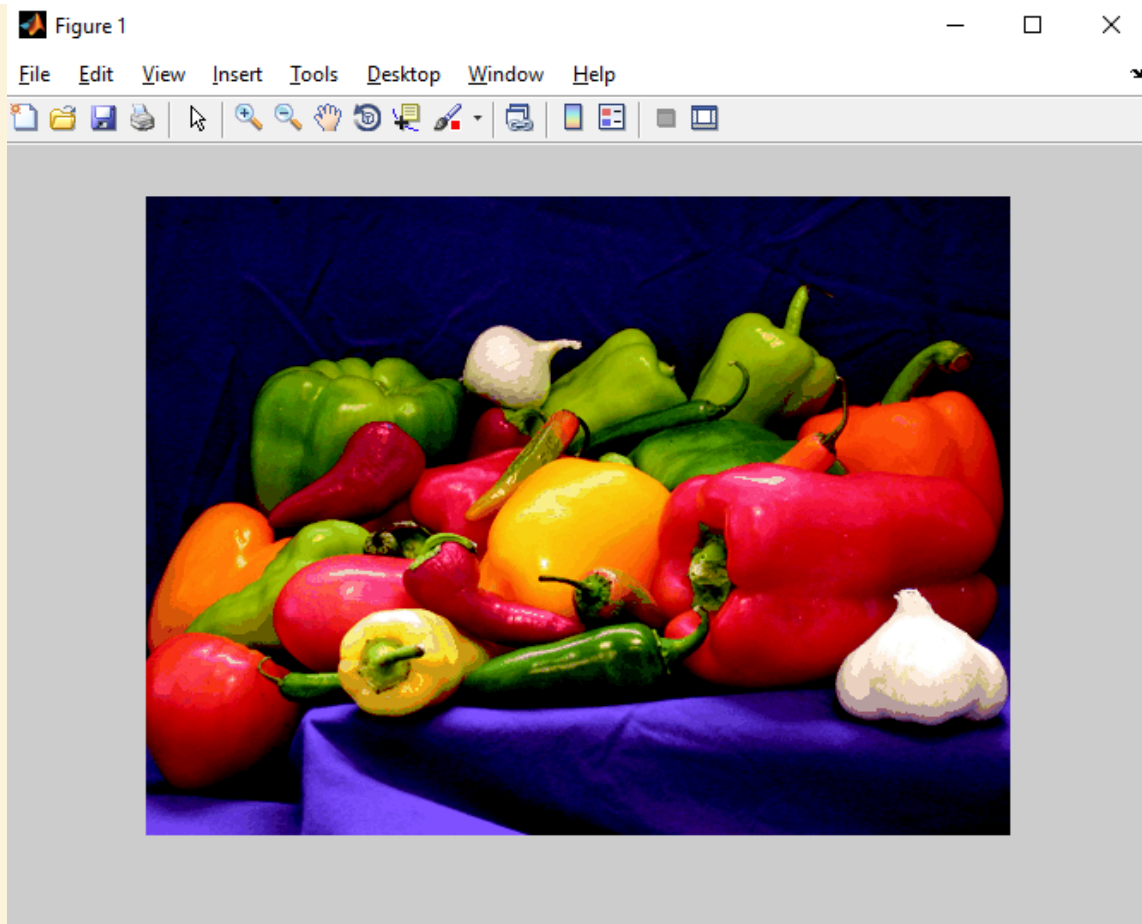
Probando lo anterior:

```
imshow(circle, [1 0 0;0 0 1])
```

Tenemos:



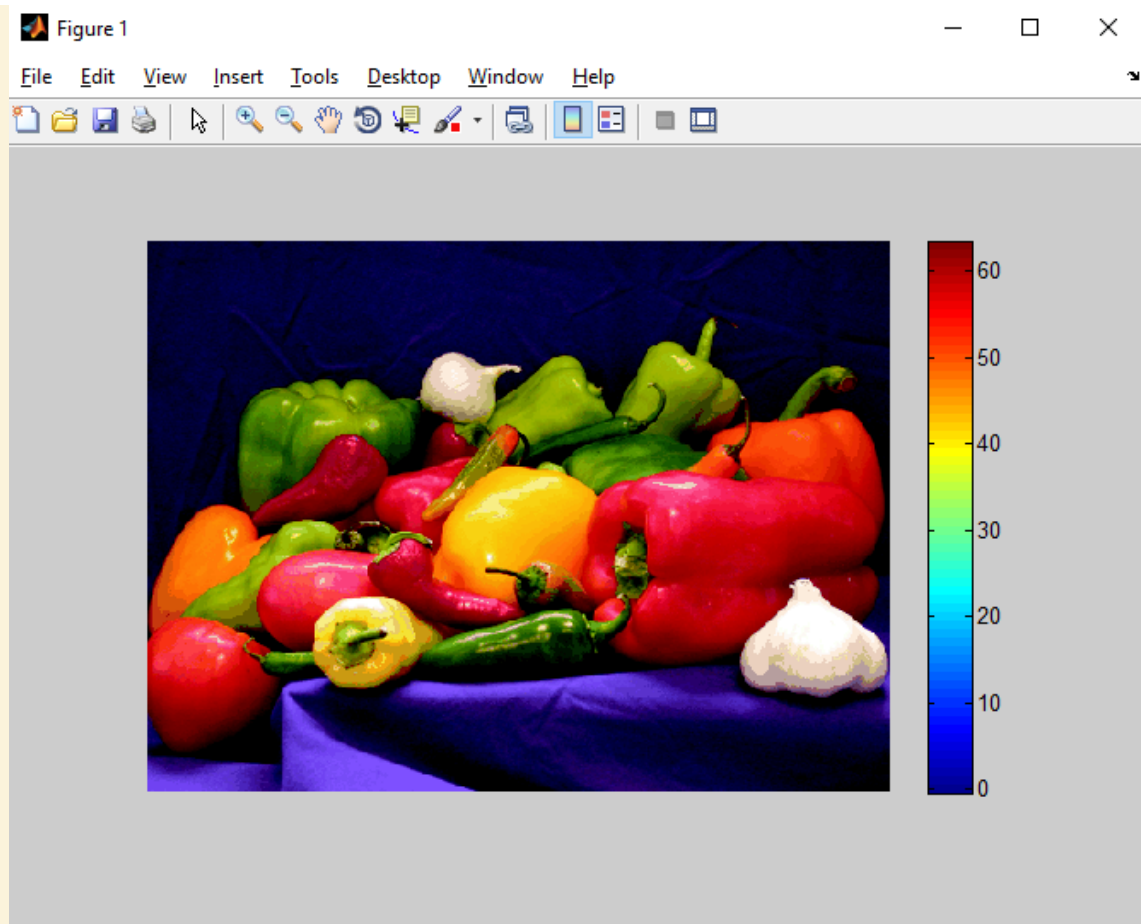
Cargando la siguiente imagen



Al poner estos comandos

```
>> imshow(pepers)
>> colorbar
>>
```

Obtenemos:



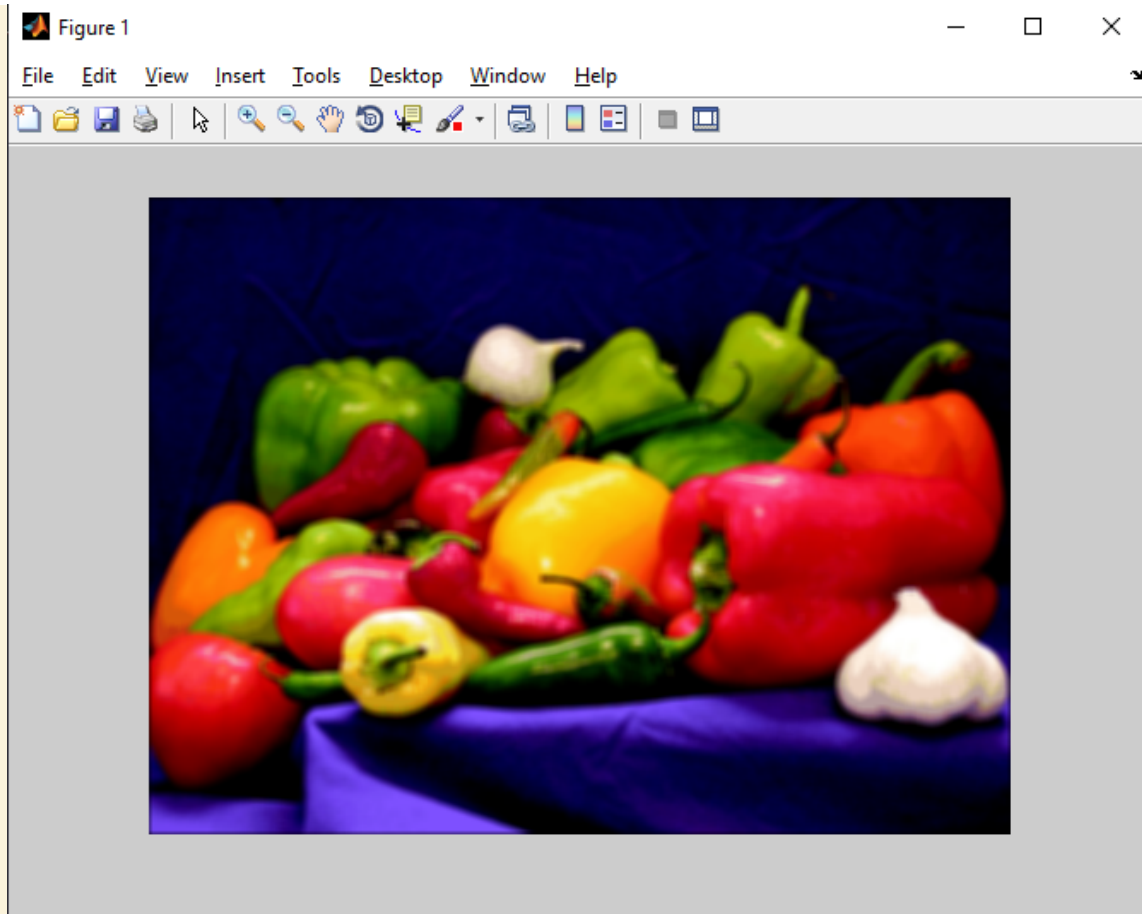
```
>> h = ones(5,5)/25
```

```
h =
```

```
0.0400    0.0400    0.0400    0.0400    0.0400  
0.0400    0.0400    0.0400    0.0400    0.0400  
0.0400    0.0400    0.0400    0.0400    0.0400  
0.0400    0.0400    0.0400    0.0400    0.0400  
0.0400    0.0400    0.0400    0.0400    0.0400
```

```
t2 = imfilter(pepers, h)
```

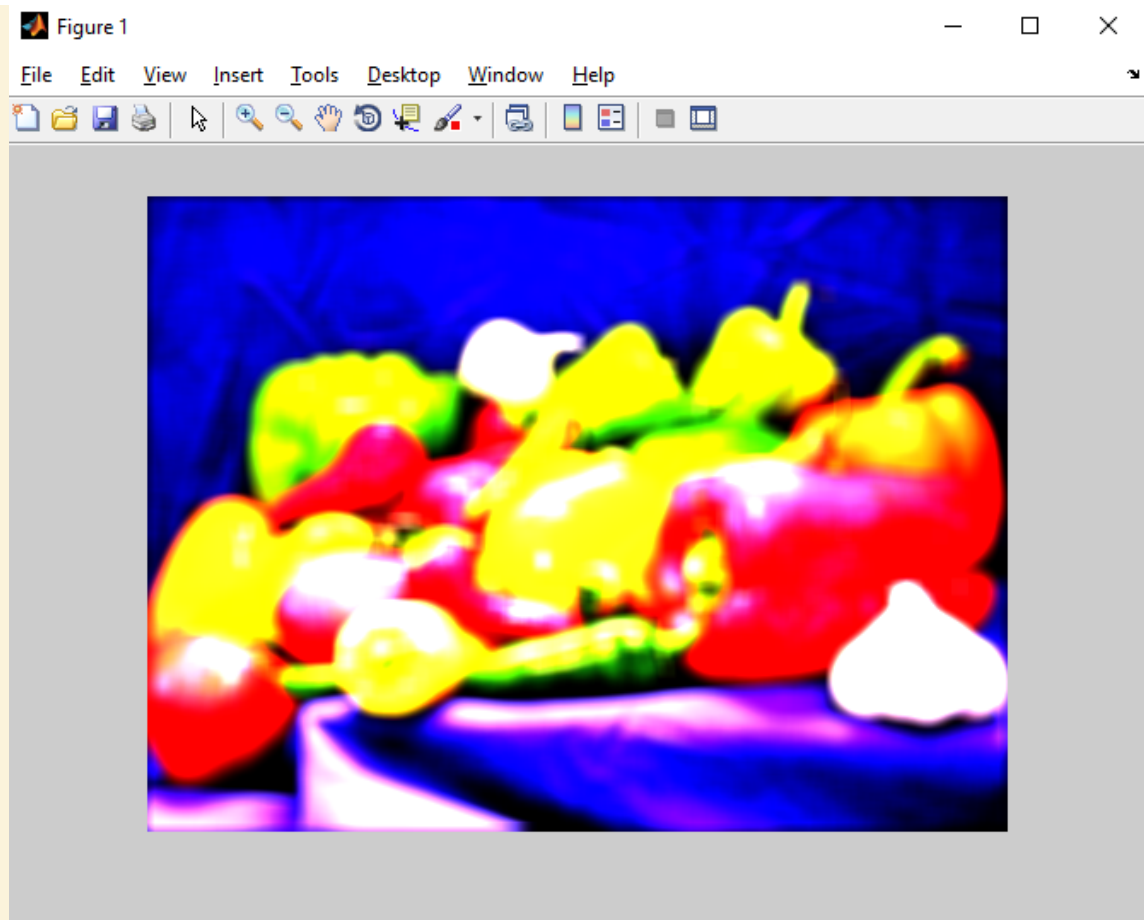
Al mostrar la nueva imagen filtrada



Ahora con estos comandos:

```
h = ones(10,10)/25  
t2 = imfilter(pepers, h)  
imshow(t2)
```

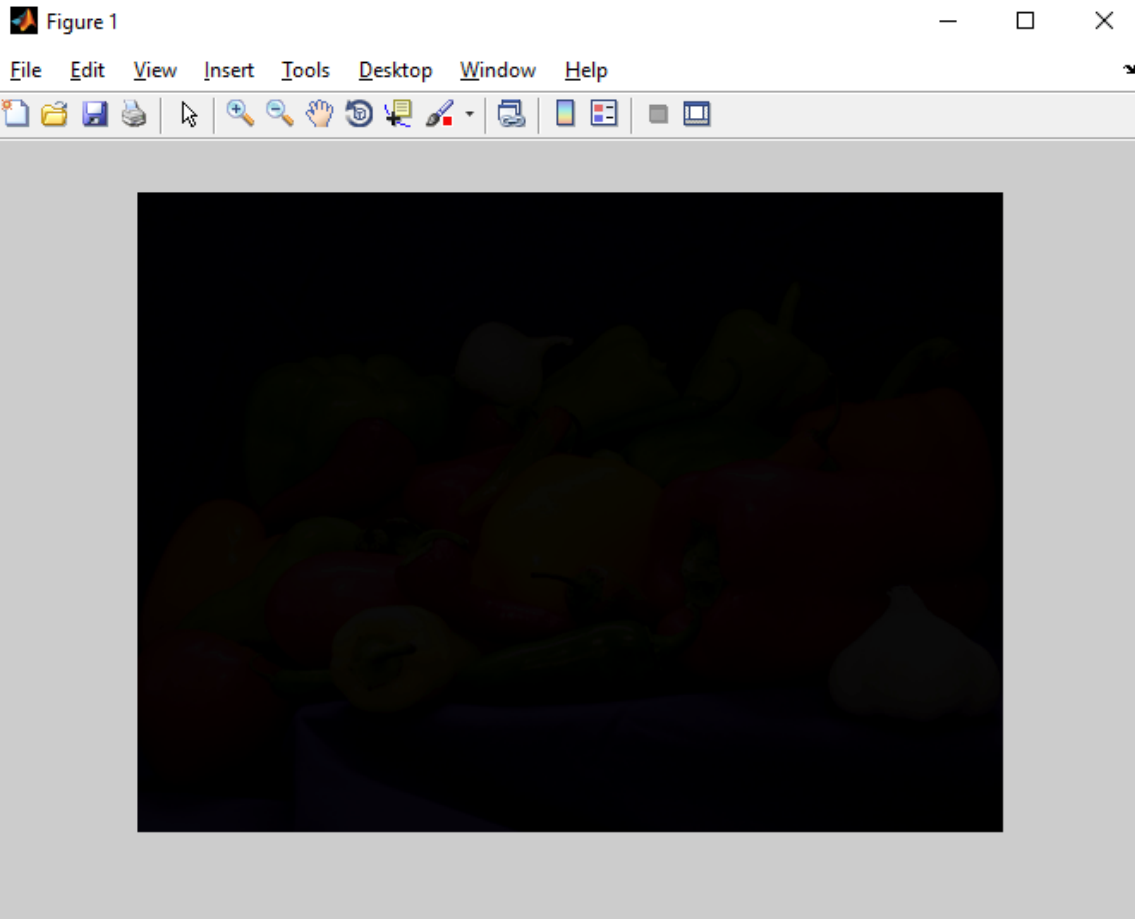
Obtenemos lo siguiente



Ahora con esta configuración:

```
h = ones(1,1)/25  
t2 = imfilter(pepers, h)  
imshow(t2)
```

Tenemos:



▼ 23-09-2021

- Los sistemas que utilizan 24 bits por píxel de pantalla pueden mostrar imágenes en color verdadero directamente, porque asignan 8 bits (256 niveles) a cada uno de los planos de color rojo, verde y azul
- En sistemas con menos colores, imshow muestra la imagen usando una combinación de aproximación de color y difuminado

Nota Si muestra una imagen en color y aparece en blanco y negro, compruebe si la imagen es una imagen indexada

- Con imágenes indexadas, debe especificar el mapa de color asociado con la imagen

Agregar barra de color a la imagen desplegada

- Este ejemplo muestra cómo mostrar una imagen en escala de grises con una barra de color que indica la asignación de valores de datos a colores
- Ver la correspondencia entre los valores de los datos y los colores mostrados mediante el uso de una barra de colores es especialmente útil si está mostrando datos de rango no convencionales como una imagen, la función es:
colorbar

Ajuste de contraste

Reasigna los valores de intensidad de la imagen al rango de visualización completo del tipo de datos


```

I = imread('pout.tif')
J = imadjust(I);
imshowpair(I,J,'montage')

```



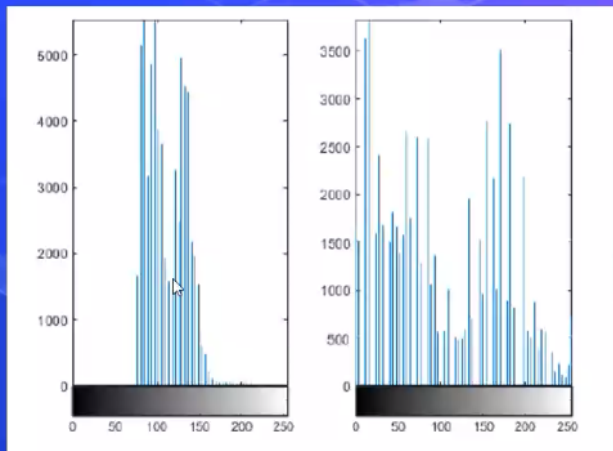
montage

- Para trazar el histograma de la imagen de ajuste
- Se debe tener en cuenta que el histograma de la imagen ajustada utiliza valores en todo el rango:

```

figure
subplot(1,2,1)
imhist(I,64)
subplot(1,2,2)
imhist(J,64)

```



utierrez Marfileño

La actividad del glosario empieza la próxima semana

▼ **24-09-2021**

No hubo clase

▼ **27-09-2021**

Instrucción de redireccionamiento o escalamiento de una imagen

Se carga la imagen en MATLAB

Si se carga bajo el nombre "f" se hace el comando:

```
g = imresize(f, 1.5)
```

f será una imagen de 100% y g será una imagen de tamaño 150%

Con el redimensionamiento, la imagen no pierde su definición

Al hacer la inversa:

```
g = imresize(f, 0.5)
```

f será una imagen de 100% y g será una imagen de tamaño 50%

Para una operación de traslación se realiza

```
j = imtranslate(g, [15, 25])
```

El punto de origen de una imagen cambiará

Ahora para una rotación se considera el centro de la imagen pero no es la única forma, pues se puede hacer en cualquier sentido y en cualquier punto de referencia

```
u = imrotate(f, ???)
```

▼ **28-09-2021**

Explicación de la práctica

Dominio de Frecuencias

Comprender que la representación de cualquier señal puede representarse en términos de la serie de Fourier

Conocer el límite de las series de Fourier para deducir la Transformada de Fourier y aplicarla al caso de imágenes

Utilizar el filtrado de imágenes para mejorarlas o restaurarlas

Serie de Fourier

Dada una función periódica de periodo 2π se busca una serie trigonométrica que la represente, es decir, que coincida con ella en algún sentido conocida como serie de Fourier

La idea que subyace en las Series de Fourier es la descomposición de una función periódica en término de señales periódicas básicas (senos y cosenos) cuyas frecuencias son múltiplos de la señal original

La serie de Fourier de una función periódica $f(x)$ de periodo T (también conocida como señal) definida en un intervalo de longitud T está dada por:

$$f(t) = \frac{a_0}{2} + \sum_{k=1}^{\infty} \left(a_k \cos \left(k \frac{2\pi}{P} t \right) + b_k \sin \left(k \frac{2\pi}{P} t \right) \right)$$

donde $a_0, a_1, \dots, a_k, \dots$ y $b_1, b_2, \dots, b_k, \dots$ son los denominados coeficientes de Fourier.

Transformada de Fourier

La transformada de Fourier es una operación matemática usada para transformar funciones entre el dominio del tiempo o espacio al dominio de la frecuencia y viceversa

El concepto de transformada de Fourier se refiere a varios conceptos de forma simultánea:

- Operación de transformación de una función
- Función resultado de la operación
- Espectro de frecuencias de una función

La función original suele recibir el nombre de $x(t)$ siendo muy común que t sea el tiempo mientras que la función transformada suele recibir el nombre de $X(f)$ en mayúscula, siendo f la frecuencia

Por ejemplo si se tiene una función $p(t)$ donde p es la potencia de una señal acústica y t el tiempo. $P(f)$ es una transformada de Fourier que informa de cómo se distribuye $p(t)$ en función de la frecuencia la potencia de la señal

En otro ejemplo, si $e(s)$ informa sobre la energía de una señal en función del espacio, $E(f)$ es una transformada de Fourier que informa sobre cómo se distribuye $e(s)$ en función de la frecuencia de la energía de la señal

Es importante destacar que aunque el tiempo, el espacio y la frecuencia son valores reales, tanto $x(t)$ y $x(s)$ como sus respectivas transformadas $X(f)$ no tienen por qué tomar valores reales

En los ejemplos de potencia y energía es posible que las magnitudes tengan **elementos complejos**

A pesar de que calcular la transformada de Fourier de una función es bastante sencillo, es necesario aprender a integrar múltiplos de funciones complejas con anterioridad

Si $x(t)$ es la función original, su transformada $X(f)$ será:

$$X(f) = \int_{-\infty}^{\infty} x(t)e^{-j2\pi ft} dt$$

Para obtener la transformada es importante tener presente varios puntos importantes:

- La integral actúa sobre t (s si la función original depende del espacio) de modo que la frecuencia f debe ser tratada como si fuese una constante, de hecho el

exponente $-j2(\pi)ft$ es una constante

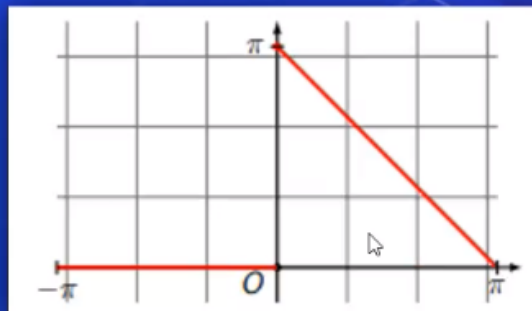
- Si la función original existe entre dos valores reales a y b, la primita de la integral se resuelve entre esos valores reales

La transformada de Fourier juega un papel muy importante en el procesamiento de imágenes, ya que es una herramienta que permite obtener la representación de información en el espacio de frecuencias y aplicando un operador en este dominio, se puede operar sobre la imagen, para detectar y realzar bordes, eliminar ruido, etc

Ejemplo de serie de Fourier

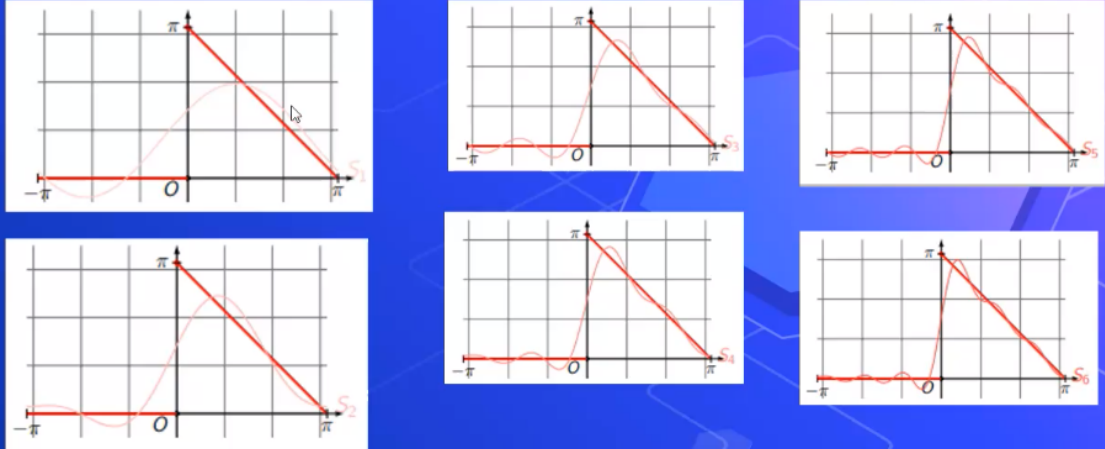
Expanda la siguiente función en una serie de Fourier:

$$f(x) = \begin{cases} 0 & \text{para } -\pi < x < 0 \\ \pi - x & \text{para } 0 \leq x < \pi \end{cases}$$



ierrez Marfileño

Elementos de la serie de expansión:



▼ 30-09-2021

La Transformada de Fourier es importante en el procesamiento de imágenes, pues permite obtener la representación de información en el espacio de frecuencias y aplica un operador en este dominio. Se puede operar sobre la imagen para detectar y realzar bordes, eliminar ruido, etc.

La transformad de Fourier es una representación de una imagen como una suma de exponenciales complejas de diferentes magnitudes, frecuencias y fases

La transformada de Fourier juega un papel fundamental en una amplia gama de aplicaciones de procesamiento de imágenes, incluida la mejora, el análisis, la restauración y la compresión

Transformada Discreta de Fourier

Suponiendo funciones bidimensionales discretas (como el caso de las imágenes digitales), la transformada discreta de Fourier o DFT se obtiene empleando sumatorias en lugar de integrales:

$$F(u, v) = \frac{1}{M \cdot N} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) \cdot e^{-j2\pi \left(\frac{ux}{M} + \frac{vy}{N} \right)}$$

Donde M representa el número de columnas (píxeles en dirección X) y N el número de filas (píxeles en dirección Y)

La transformación inversa viene dada por la expresión:

$$f(x, y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) \cdot e^{j2\pi \left(\frac{ux}{M} + \frac{vy}{N} \right)}$$

El espectro de Fourier, la fase y el espectro de energía se calculan de la misma forma que en el caso de una función continua, con la única diferencia de que las variables independientes son ahora discretas

El componente de fase es relevante en cuanto a la estructura espacial de la imagen y contiene información sobre la posición relativa de las características u objetos en la imagen

Propiedades de la transformada discreta:

- La separabilidad o posibilidad de escribir la transformada de la forma (suponiendo una imagen cuadrada de dimensión NxN):

$$F(u, v) = \frac{1}{N} \sum_{x=0}^{N-1} e^{-j\frac{2\pi ux}{N}} \cdot \sum_{y=0}^{N-1} f(x, y) \cdot e^{-j\frac{2\pi vy}{N}}$$

Cuya principal ventaja es que, tanto $f(x, y)$ como $F(u, v)$ pueden obtenerse en dos pasos sucesivos, aplicando la transformada de Fourier unidimensional o su inversa para cada fila y después para cada columna

- La traslación o desplazamiento de la función $f(x, y)$ no afecta la magnitud de su transformada de Fourier. Es interesante debido a que el análisis visual de la transformada suele limitarse al estudio de su magnitud o módulo
- Periodicidad: La DFT tiene un periodo de longitud N. La magnitud de la transformada está centrada en el origen. Para visualizar un periodo completo basta con desplazar el origen de la transformada al centro de la imagen ($U = N/2$, $V = N/2$). De esta forma, al trasladar $F(0, 0)$ al centro de una imagen, el

píxel central en la imagen de la transformada de Fourier representa el valor de la intensidad media en la imagen

- La rotación de $f(x,y)$ en un ángulo θ genera una rotación de $F(u,v)$ en el mismo ángulo y viceversa
- La transformada y su inversa son distributivas en la suma, pero no en la multiplicación

$$\mathfrak{F}[f_1(x,y) + f_2(x,y)] = \mathfrak{F}[f_1(x,y)] + \mathfrak{F}[f_2(x,y)]$$

Sin embargo:

$$\mathfrak{F}[f_1(x,y) \cdot f_2(x,y)] \neq \mathfrak{F}[f_1(x,y)] \cdot \mathfrak{F}[f_2(x,y)]$$

- Valor medio: El primer componente de la transformada $F(0,0)$ representa el valor medio de la función discreta original tal y como se ha comentado anteriormente

▼ 01-10-2021

No hubo clase

▼ 04-10-2021

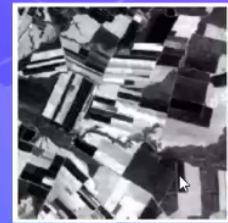
Comprender que la representación de cualquier señal puede representarse en términos de la serie de Fourier
Conocer el límite de las series de Fourier para deducir la Transformada de Fourier y aplicarla al caso de imágenes
Utilizar el filtrado de imágenes para mejorarlas o restaurarlas

Los valores de los píxeles en la imagen de la transformada de Fourier, $F(u,v)$, representan los componentes de frecuencia espacial, verticales y horizontales, de la imagen original, análogamente a la frecuencia de una señal en el tiempo. Así, una señal sinusoidal con alta frecuencia, por ejemplo, varía muy rápidamente, mientras que otra con una frecuencia baja varía lentamente en el tiempo. De la misma forma, una imagen con una elevada frecuencia espacial en la dirección horizontal, contiene cambios frecuentes de intensidad en la dirección horizontal.

En las siguientes figuras se muestran ejemplos de tres imágenes con sus respectivas transformadas.

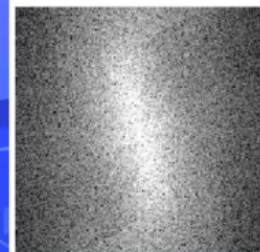
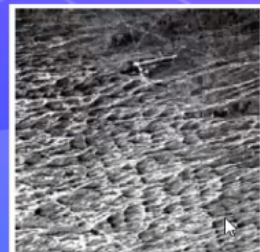
En ellas se pueden ver tres distribuciones espaciales distintas y el efecto que tienen sobre la representación en forma de imagen de sus transformadas, incluyendo el claro componente direccional de las frecuencias.

La primera de ellas representa una escena agrícola con profusión de pequeñas parcelas de distintos cultivos, lo que conlleva a una gran cantidad de altas frecuencias en varias direcciones, generando una apariencia de estrella en su transformada, cuyos brazos son perpendiculares a las orientaciones dominantes.

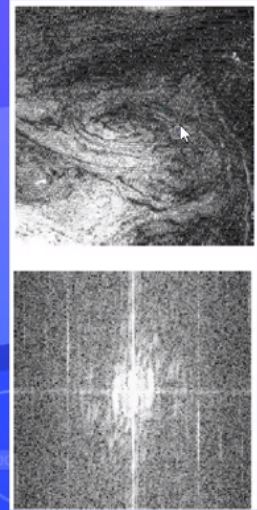


La segunda imagen representa un paisaje dunar, con una orientación dominante (NE-SW) debida a la acción de los vientos en la zona.

Su transformada de Fourier se muestra en forma de nube, debido a que existe poca definición en la orientación dominante, alargada según la dirección perpendicular a la orientación de las dunas, es decir, según la dirección con mayor variación de la frecuencia espacial.



Por último, la tercera imagen representa una zona oceánica en la que se aprecian corrientes en forma helicoidal, lo cual se transforma en el espacio de Fourier, en una serie de pequeñas manchas próximas al origen (bajas frecuencias) y poco definidas linealmente (falta de periodicidad). Nótese que, en todas ellas, existe una línea vertical de elevada amplitud que pasa por el origen, debida al patrón periódico generado por el efecto de barrido transversal de los sensores, el cual tan solo se aprecia visualmente en la imagen inferior

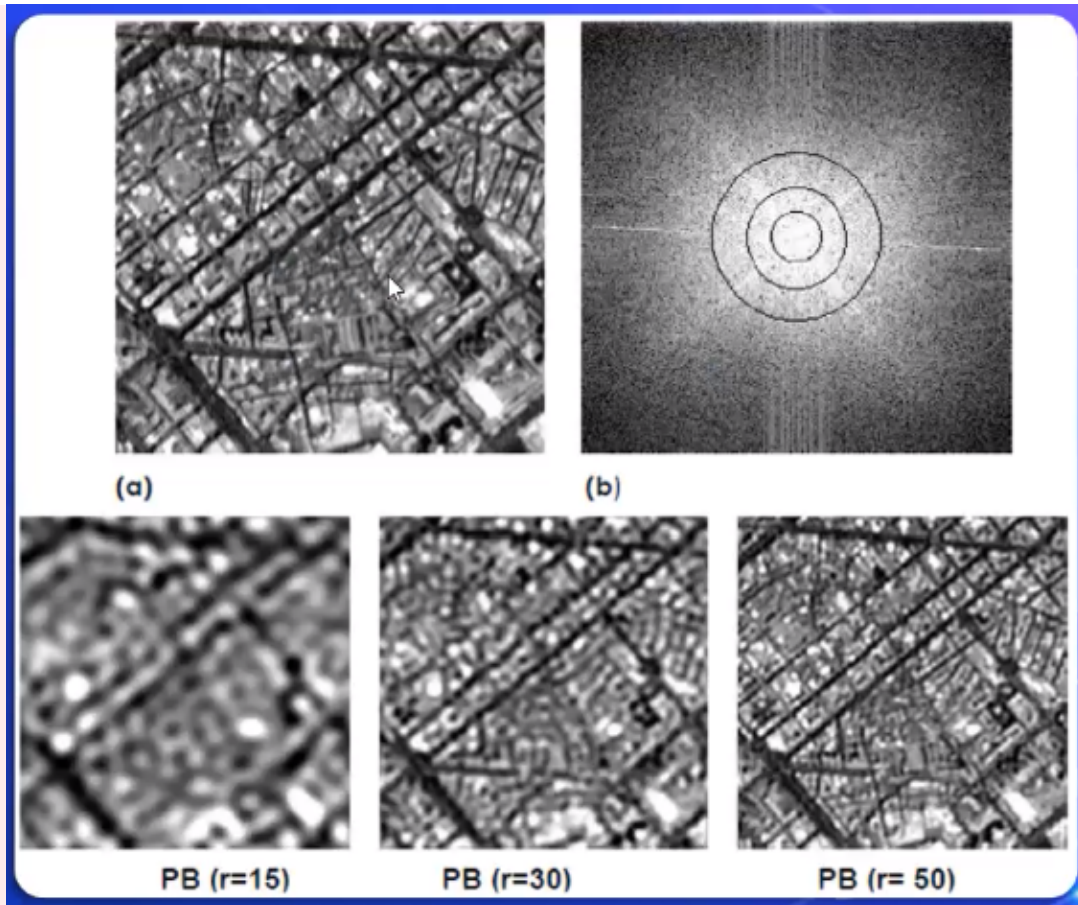


Un filtro **paso-bajo** consiste en la eliminación o atenuación de las altas frecuencias de la imagen

Para ello, bastará definir una máscara binaria de forma que al aplicarla sobre el espectro de frecuencias mantenga los valores bajos, situados en el centro de la imagen, y elimine los altos

Esto se consigue mediante círculos binarios de valor uno en el interior y cero en el exterior

Cuanto menor sea el radio del círculo, mayor será el efecto del filtro, ya que el intervalo de altas frecuencias que se elimina en la multiplicación ($F(u,v) \times H(u,v)$) será mayor

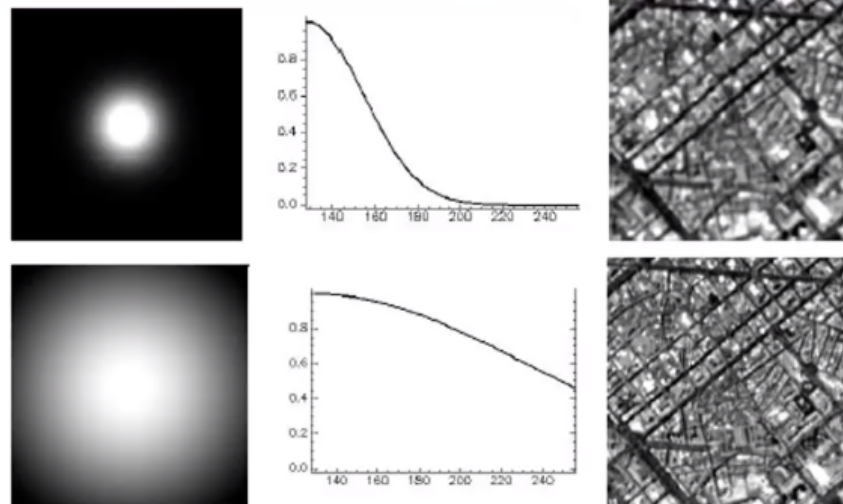


Al contrario, cuanto mayor sea el radio del círculo, menor es el efecto del filtro

También puede determinarse la *discriminación* del filtro, entendida como la suavidad con la que pasa de la zona de corte a la de paso. Así, si en vez de definirse un círculo binario, el filtro se diseña como una función gaussiana con valores de 0 a 1, la elección de la desviación típica de la misma permite controlar la pendiente del corte de las frecuencias del filtro.

En la siguiente figura se muestran dos ejemplos, uno con una función gaussiana con $\sigma=0,2$, y en el otro con $\sigma=0,8$.

Tal y como se aprecia en los gráficos de los perfiles de los filtros, el primero realiza un filtrado de las altas frecuencias mucho más severo que el segundo.

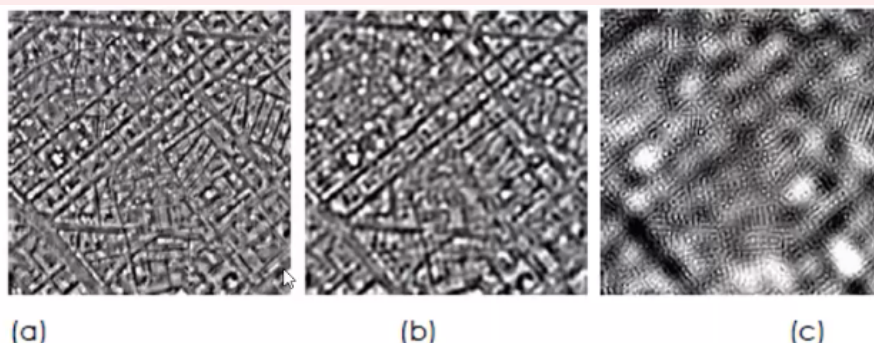


- Izquierda: Funciones de transferencia de dos filtros gaussianos, $\sigma=0,2$ (arriba) y $\sigma=0,8$ (abajo). Centro: Semiperfiles de ambos filtros. Derecha: Resultados de las imágenes filtradas.

Análogamente, un filtro **paso-alto** podrá diseñarse de forma sencilla mediante una máscara binaria circular, tal que dentro del círculo tenga como valor cero, y fuera del mismo la unidad. Cuanto mayor sea el radio del círculo, mayor será el intervalo de bajas frecuencias que se elimina y, por lo tanto, más severo será el efecto del filtrado, al contrario de los que ocurría en el diseño de filtros paso-bajo.

Si el filtro se define mediante una máscara binaria en forma de anillo, con un radio interno y otro externo, el efecto que se consigue es el filtrado de las frecuencias intermedias o extremas.

Estos filtros se denominan **paso-banda**.



Imágenes resultantes de aplicar diferentes filtros en el espacio de las frecuencias: (a) filtro paso-alto circular de radio=15; (b) paso-banda circular de frecuencias intermedias (entre 10 y 40); y (c) el mismo paso-banda circular pero de frecuencias extremas.

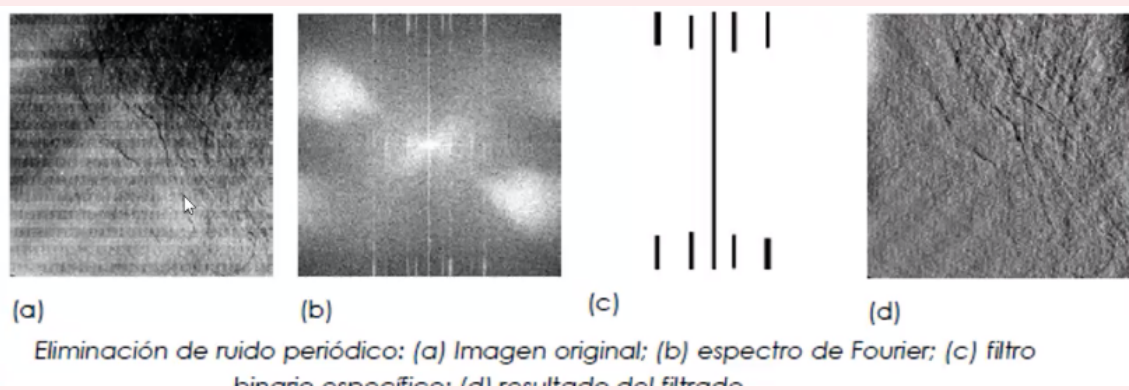
El **diseño de filtros específicos** en el espacio de Fourier es especialmente relevante cuando se trata de eliminar el **ruido periódico** en la imagen

Por ejemplo, supongamos la imagen de la siguiente figura en la que se representa un detalle del mar Mediterráneo

En ella se aprecia claramente el bandeo horizontal provocado por el sucesivo barrido transversal del sensor TM de Landsat

El espectro de frecuencias refleja este efecto en forma de líneas verticales con diferentes frecuencias

El diseño de un filtro binario que enmascare o elimine estas líneas, y su multiplicación sobre el espectro, permite obtener la imagen filtrada en la que el patrón de bandeo horizontal o ruido periódico ha desaparecido



Nótese además cómo en el resultado aparecen ligeramente más realzadas las altas frecuencias que en la imagen original, debido a que el filtro elimina una pequeña parte de las bajas frecuencias correspondientes a la zona central (origen) del espectro

▼ 05-10-2021

No hubo clase

▼ 06-10-2021

No hubo clase por el Congreso de Ciencias Exactas

▼ 07-10-2021

No hubo clase por el Congreso de Ciencias Exactas

▼ **08-10-2021**

No hubo clase por el Congreso de Ciencias Exactas

▼ **11-10-2021**

Uso de la imagen del tigre para más ejemplos de manipulación de imágenes

▼ **12-10-2021**

Filtro gaussiano

El **filtro gaussiano** es un filtro de suavizado lineal que puede eliminar el ruido gaussiano y se usa ampliamente en el proceso de eliminación de ruido del procesamiento de imágenes

El filtrado gaussiano es el proceso de promedio ponderado de toda la imagen

El valor de cada píxel se obtiene mediante el promedio ponderado de sí mismo y otros valores de píxeles en la vecindad

La operación específica del filtrado gaussiano es usar una plantilla (o convolución, máscara) para escanear cada píxel en la imagen y usar la escala de grises promedio ponderada de los píxeles en la vecindad determinada por la plantilla para reemplazar el valor del píxel central de la plantilla

Ruido gaussiano

Entre los diferentes tipos de ruido en imágenes, el más común es el **ruido gaussiano**, que aparece en el momento de la adquisición de la imagen por un dispositivo en malas condiciones, mala iluminación o altas temperaturas y tiene como consecuencia el emborronamiento de todos los píxeles de la imagen

En el ruido de tipo Gaussiano, la intensidad de todos y cada uno de los píxeles que componen la imagen se ven afectados y cambian su valor, de acuerdo con una distribución normal

Su función de densidad de probabilidad $p_q(x)$ viene dada en términos de la media μ y la varianza σ^2 de una variable aleatoria x

Se expresa en la siguiente ecuación:

$$p_q(x) = (2\pi\sigma^2)^{-1/2} e^{-(x-\mu)^2/2\sigma^2}$$

El efecto visual de la imagen generada por la tecnología de desenfoque gaussiano es como observar la imagen a través de una pantalla semitransparente, lo cual es obvio por el efecto de imagen desenfocado de la lente y la Efecto a la sombra de la iluminación ordinaria diferente

El suavizado gaussiano también se utiliza en la etapa de preprocesamiento de los algoritmos de visión por computadora para mejorar el efecto de imagen de las imágenes a diferentes escalas

Desde un punto de vista matemático, el proceso de desenfoque gaussiano de una imagen es la convolución de la imagen y la distribución normal

Dado que la distribución normal también se denomina distribución gaussiana, esta técnica se denomina desenfoque gaussiano

Filtro gaussiano



Se abre actividad de la actividad de glosario hasta el segundo parcial

Mi entrada en el glosario

`imread`

Sintaxis: `variable = imread('ruta_de_la_imagen')`

Descripción: Esta función carga una imagen dentro de una variable de MATLAB para poder usarse posteriormente con otras funciones.

Opciones: La función pide que se ingrese dentro de los paréntesis entre comillas simples la ruta del archivo, que podrá ser la ruta completa o una ruta relativa a la del archivo `.mat` que se esté utilizando.

Finalmente, los formatos soportados por MATLAB son TIFF, JPEG, GIF, BMP, PNG y XWD

Ejemplos:

```
imagen = imread('imagen.jpg')
```

```
img = imread('C:\imagenes\imagen.png')
```

Nombre: Joel Alejandro Espinoza Sánchez

`imread`

Sintaxis: `variable = imread('ruta_de_la_imagen')`

Descripción: Esta función carga una imagen dentro de una variable de MATLAB para poder usarse posteriormente con otras funciones.

Opciones: La función pide que se ingrese dentro de los paréntesis la ruta del archivo entre comillas simples. La ruta podrá ser la ruta completa o una ruta relativa a la del archivo .mat que se esté utilizando.

Finalmente, los formatos soportados por MATLAB son TIFF, JPEG, GIF, BMP, PNG y XWD

Ejemplos:

```
imagen = imread('imagen.jpg')
```

```
image = imread('imagenes\imagen.jpg')
```

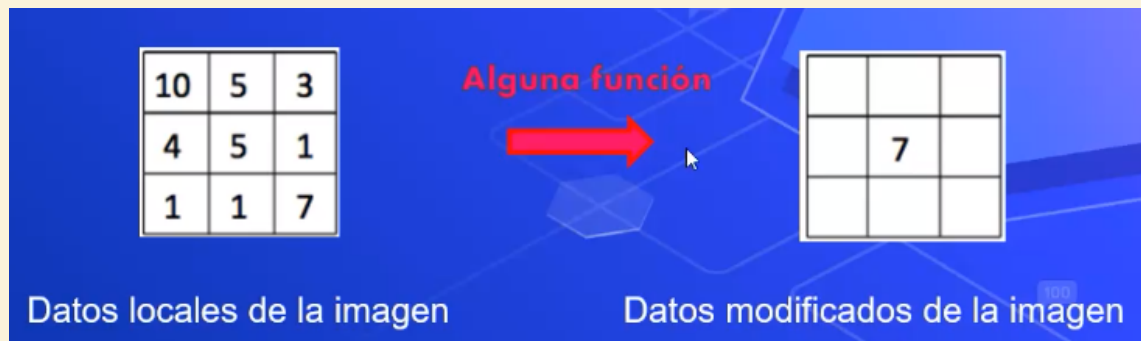
```
img = imread('C:\imagenes\imagen.png')
```

Nombre: Joel Alejandro Espinoza Sánchez

▼ 13-10-2021

Convolución

Es la operación de modificar los pixeles de una imagen basándose en alguna función sobre el vecindario local de cada pixel



Características de la convolución

- Es una operación lineal
- Es la operación más general que se puede aplicar a una imagen

- A veces también nos referimos a la convolución como filtrado
- Filtrado espacial: Directamente aplicable mediante una máscara
- Las máscaras de convolución o filtros, no suelen ser grandes (3x3 o 5x5) para ahorrar tiempo de cómputo

Definición general de la convolución

$$f(x) * g(x) = \int_{-\infty}^{\infty} f(u) g(x-u) du$$

Convolución para imágenes digitales

$$F(x, y) = f(x, y) * g(x, y) = \sum_i \sum_j f(i, j) g(x-i, y-j)$$

Donde g es la máscara de convolución espacial

Propiedades de la convolución

Sean a , b , c y d imágenes continuas o discretas

- La convolución es conmutativa

$$c = a \otimes b = b \otimes a$$

- La convolución es asociativa

$$c = a \otimes (b \otimes c) = (a \otimes b) \otimes c = a \otimes b \otimes c$$

- La convolución es distributiva

$$c = a \otimes (b + d) = (a \otimes b) + (a \otimes d)$$

La convolución provee una manera de multiplicar dos arreglos de números, generalmente de tamaños diferentes pero de la misma dimensionalidad, para

producir un tercer arreglo de números de la misma dimensionalidad

Esto puede ser utilizado para implementar operadores cuyos valores de píxeles de salida sean simples combinaciones lineales de ciertos valores de píxeles de entrada

Uno de los arreglos de entrada es usualmente una imagen en niveles de gris en 2D

El segundo arreglo es usualmente mucho más pequeño y también es 2D (aunque puede ser de 1x1, i.e. un pixel)

I_{11}	I_{12}	I_{13}	I_{14}	I_{15}	I_{16}	I_{17}	I_{18}	I_{19}
I_{21}	I_{22}	I_{23}	I_{24}	I_{25}	I_{26}	I_{27}	I_{28}	I_{29}
I_{31}	I_{32}	I_{33}	I_{34}	I_{35}	I_{36}	I_{37}	I_{38}	I_{39}
I_{41}	I_{42}	I_{43}	I_{44}	I_{45}	I_{46}	I_{47}	I_{48}	I_{49}
I_{51}	I_{52}	I_{53}	I_{54}	I_{55}	I_{56}	I_{57}	I_{58}	I_{59}
I_{61}	I_{62}	I_{63}	I_{64}	I_{65}	I_{66}	I_{67}	I_{68}	I_{69}

K_{11}	K_{12}	K_{13}
K_{21}	K_{22}	K_{23}

Una máscara es una matriz (usualmente) más pequeña

Máscaras de diferentes tamaños conteniendo diferentes patrones de números dan resultados diferentes

1	1	1
1	1	1
1	1	1

Set of coordinate points =

{ (-1, -1), (0, -1), (1, -1),
(-1, 0), (0, 0), (1, 0),
(-1, 1), (0, 1), (1, 1) }

La implementación se hace deslizando la máscara sobre la imagen, generalmente comenzando desde la esquina superior izquierda y moviéndose a todas las posiciones donde quede completamente dentro de los límites de la imagen

Si la imagen es de $M \times N$ y la máscara de $m \times n$ el tamaño de salida será $(M-m+1) \times (N-n+1)$

Cada posición de la máscara corresponde a un solo pixel de salida calculado por la función:

$$O(i, j) = \sum_{k=1}^m \sum_{l=1}^n I(i+k-1, j+l-1) K(k, l)$$

donde $i = 1..M-m+1, j = 1..N-n+1$

▼ 14-10-2021

No hubo clase

▼ 15-10-2021

No hubo clase

▼ 18-10-2021

Se nos platicó de un conjunto de librerías para el nuevo tratamiento fuera del conjunto de imágenes

▼ 19-10-2021

Ruido

Cuando se quiere procesar una imagen habrán ocasiones en las que las imágenes haya ruido

Las imágenes pueden tener cierto ruido por convertir una imagen óptica a una señal eléctrica continua que posteriormente es muestreada siempre generará un ruido en la imagen digital

El otro es el ruido sal y pimienta que es gracias a la transmisión de canales digitales

El ruido es información no deseada en una imagen que causa contaminación

Se puede notar en una imagen como píxeles aislados con un nivel de gris distinto a los píxeles que lo rodeen (también llamados vecinos)

Hay dos puntos claves que causan ruido:

1. La fuente que genera la imagen
2. El medio en que se transmite la señal eléctrica generada

Tipos de ruido

Dentro de los tipos existe el ruido gaussiano que es cómo se distribuye a lo largo de la imagen

La imagen se distorsionaría del centro hacia las orillas

Operaciones de suavizado

Este tipo de operaciones disminuye el ruido de una imagen por errores al momento de procesarla

Algunas eliminan el ruido como a continuación

- **Filtrado de una imagen:** Proceso de modificar una imagen para mejorarla eliminando ruidos o elementos no deseados. Analizando los píxeles que hay en un vecindario y dependiendo del algoritmo y la acción deseada (atenuar, eliminar o resaltar) definirá el valor de los píxeles del vecindario
- **Filtro Gaussiano:** Suaviza la imagen eliminando el ruido pero provoca un efecto de pérdida de nitidez y causa efecto borroso

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

La máscara da mayor peso al pixel central y a los píxeles que se encuentran cercano a este y menor a los más alejados

La forma de generar un filtro gaussiano en MATLAB es:

```
Nombre_Filtro = fspecial('gaussian')
```

Para aplicarle filtro a la imagen se establece una variable de la siguiente forma:

```
Nueva_image = imfilter(imagen_con_ruido, Nombre_Filtro)
```

Para agregar ruido se puede introducir con el siguiente código:


```
Nuema_image = imnoise(imagen_original, 'tipo_ruido', densidad)
```

Implementamos la solución a MATLAB:

```
% Ejercicio Filtro Gaussiano
original = imread('Oceano1.jpg') % Lee la imagen y la asigna a original
ruido = imnoise(original, 'gaussian') % Le pusimos ruido a la imagen original
gauss = fspecial('gaussian') % Matriz para transformar la imagen
filtrado_gauss = imfilter(ruido, gauss) % Imagen corregida con el filtro gauss

% Representaciones de las imagenes
subplot(2,2,1), subimage(original), title('Imagen original');
subplot(2,2,2), subimage(ruido), title('Imagen con ruido');
subplot(2,2,3), subimage(filtrado_gauss), title('Imagen con filtro de Gauss')
```

MATLAB tiene un filtro general para el ruido Gaussiano, el anterior es una convolución general. Unas opciones pueden ser:

```
Fun_fil = immgaussfilt(ruido)
Fun_fil = immgaussfilt(ruido, sigma)
```

Útil en radiografías

▼ 20-10-2021

Generación de imágenes

Es posible generar imágenes binarias a partir de la definición de ceros y unos en matrices

Dependiendo de si se desea crear un fondo con ceros o unos se generan las dimensiones de la imagen a partir de las dimensiones de la matriz

Luego es posible incluir objetos geométricos con el valor de contraste

Es decir si se desea un fondo en negro la matriz inicial es a partir de ceros y los objetos en unos (blanco)

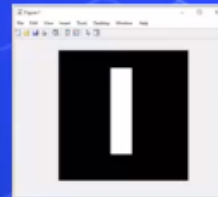
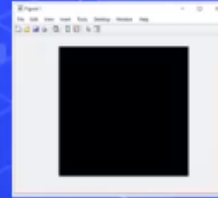
Por el contrario si se desea un fondo en blanco la matriz inicial es a partir de unos y los objetos en ceros (negro)

Ejemplo

Crear una imagen binario con fondo en negro de 30x30 pixeles y un objeto rectangular ubicado en las coordenadas (5,24),(13:17) en blanco:

```
f = zeros(30,30)
```

```
f(5:24,13:17) = 1
```

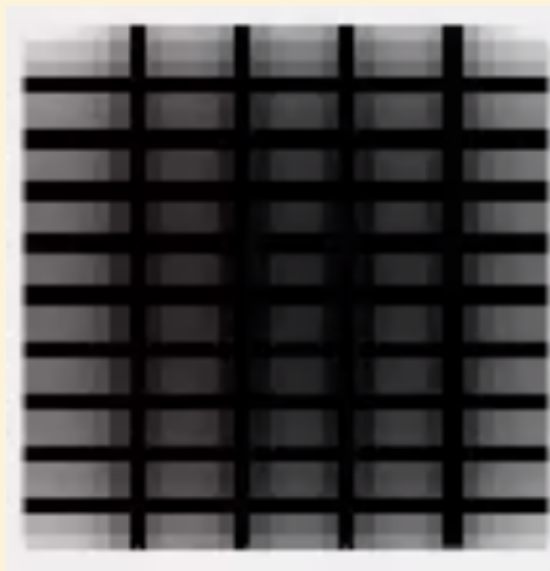


Usando la transformada de Fourier para dos dimensiones:

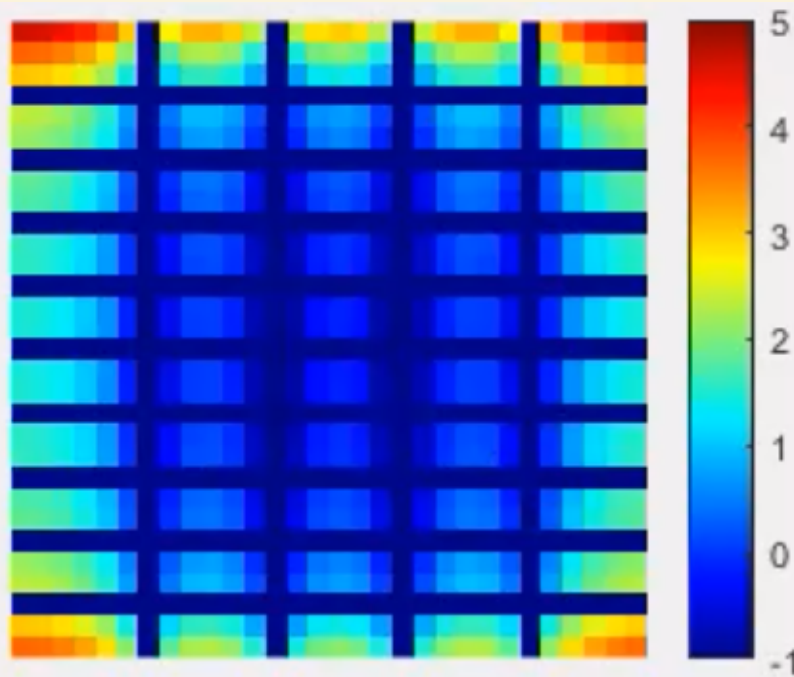
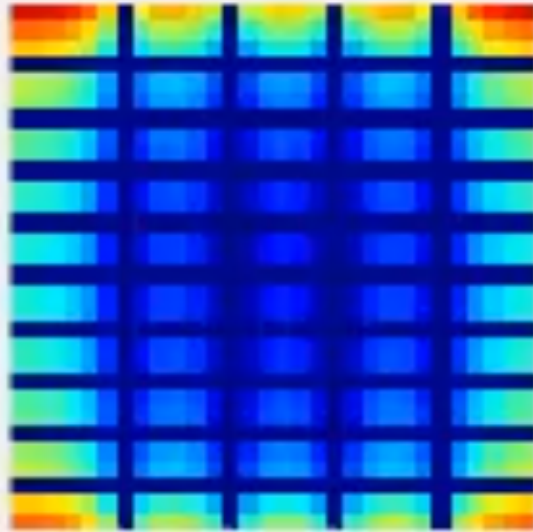
```
>> F = fft2(f)
```

Se ampliará con la función

```
>> imshow(F2, [-1, 5], 'InitialMagnification', 'fit')
```



```
>> colormap(jet)
>>
```



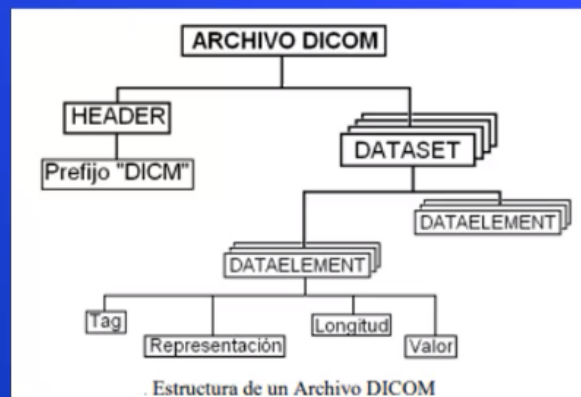
▼ 21-10-2021

DICOM (*Digital Imaging and Communications in Medicine*) es un protocolo estándar de comunicación entre sistemas de información y a la vez un **formato** de almacenamiento de imágenes médicas que aparece como solución a los problemas de interoperabilidad entre tipos de dispositivos

Una imagen médica por sí misma no aporta suficiente información

Los formatos existentes de imágenes digitales se quedan cortos para las imágenes medicas

En DICOM se interpreta el **mundo real** como todos los datos físicos o descriptivos como por ejemplo el nombre del paciente, el tipo de estudio, el dispositivo médico, los parámetros de la adquisición, la imagen digital, etc. que son vistos por DICOM como elementos con sus respectivos atributos y propiedades

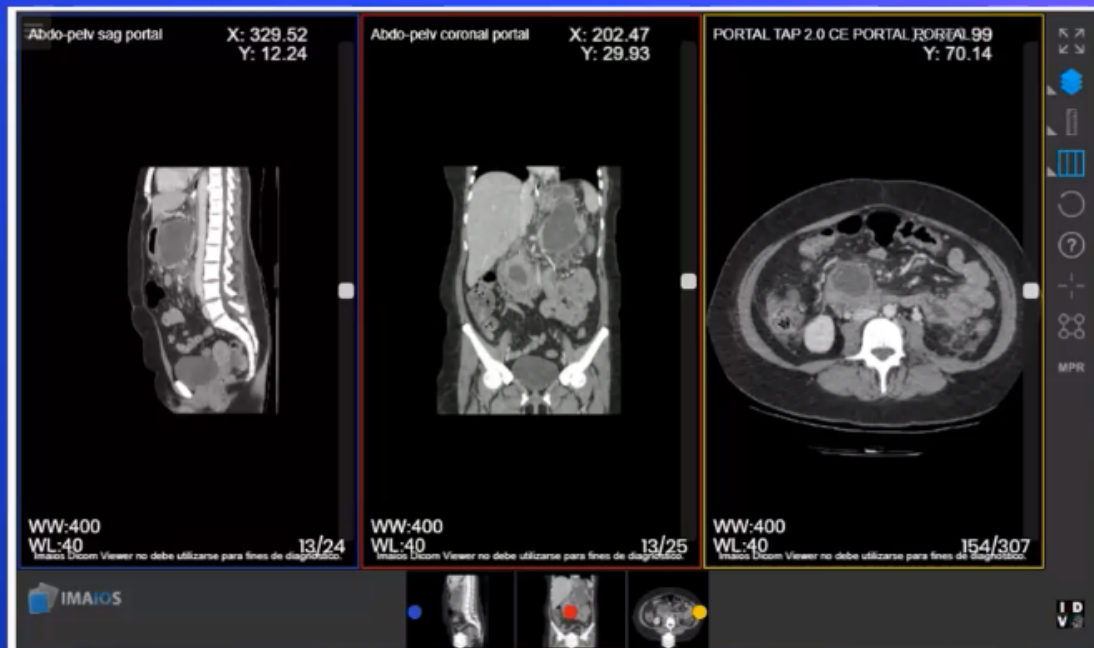


Por lo general un archivo DICOM es reconocible por su extensión *.dcm, sin embargo esto no es una exigencia del Estándar, por lo que la forma de diferenciarlo es por medio del HEADER o cabecera que consta de 128 bytes de archivos de preámbulo y 4 bytes de prefijo DICM. El preámbulo puede estar en blanco o contener información sobre la aplicación principal con la que debe ser ejecutado.

TAG	Descripción	Tipo
(0028,0002)	Samples per Pixel	Int
(0028,0008)	Number of Frames	Int
(0028,0010)	Rows	Int
(0028,0011)	Columns	Int
(0028,0100)	Bits Allocated	Int
(0028,0101)	Bits Stored	Int
(0028,0102)	High Bit	Int
(0028,0103)	Pixel Representation	Int
(0028,1050)	Window Center	Int
(0028,1051)	Window Width	Int
(0028,1052)	Rescale Intercept	Int
(0028,1053)	Rescale Slope	Int
(7FE0,0010)	PixelData	Byte[] ó Uint16[]

Data Elements Necesarios para la lectura adecuada de una imagen

Visor DICOM



DICOM. Historia



- Creado por ACR (American College of Radiology) y NEMA (National Electrical Manufacturers Association)
- ACR-NEMA 1.0 y 2.0 en 1985-1988
- DICOM (3.0) inicial en 1993
- Desarrollo continuo de DICOM 3.0 desde entonces
- Desarrollo conjunto con HL7 desde 1995

Poco éxito

Estándar real

Imágenes médicas

Ejemplos

- ⬡ Placa de rayos X
- ⬡ Tomografía Computada
- ⬡ Ecografía obstétrica
- ⬡ Resonancia magnética
- ⬡ Franjas Moiré
- ⬡ Tomografía por emisión de positrones

Imágenes Médicas

Finalidad de las imágenes

- ⬡ **Estructura interna a evaluar (huesos)**
- ⬡ **Función vital a estudiar**
- ⬡ **Documentación de situación biológica**
- ⬡ **Resumen visual de información compleja**
- ⬡ **Integración de imágenes diferentes**
- ⬡ **Ayuda para acciones (cirujía, cateterismo, etc.)**

Imágenes Médicas

Particularidades respecto a otras imágenes

- **Dificultad de acceso al objeto**
- **Variabilidad biológica**
- **Evolución en el tiempo**
- **Necesidad de interpretación experta**
- **Complejidad**
- **Preservar la privacidad del paciente**
- **Gravedad de consecuencias de artefactos**

IMÁGENES MÉDICAS

Noción de la diversidad de principios físicos que permiten construir y obtener imágenes para uso médico

Manejar los principios de algunas modalidades de imágenes

Conocer los elementos constitutivos de la instrumentación

Adquirir práctica de manejo de imágenes en ámbito telemático (DICOM)

Proyecto de interconexión de equipos de imágenes y redes para uso en Hospitales

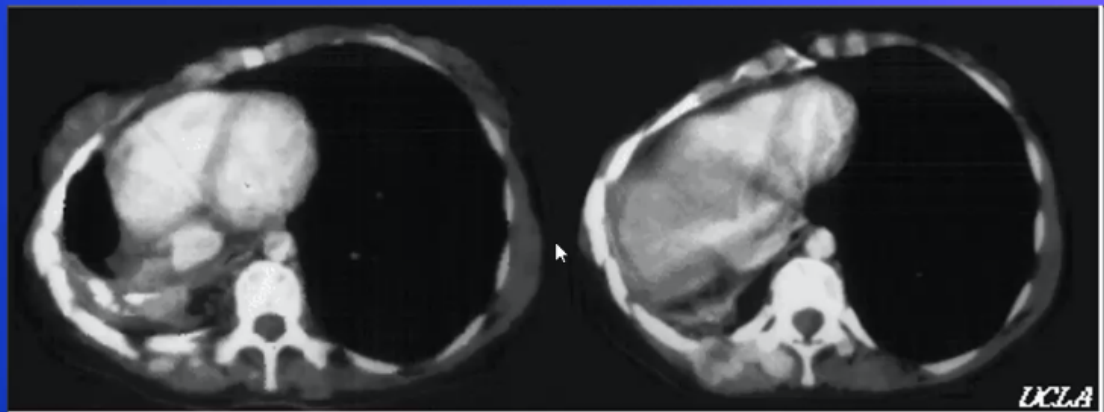
Precauciones para pacientes y operadores, blindajes, envergadura y dimensiones del mantenimiento

Todos estos conceptos cuantificados (tarea del ingeniero)

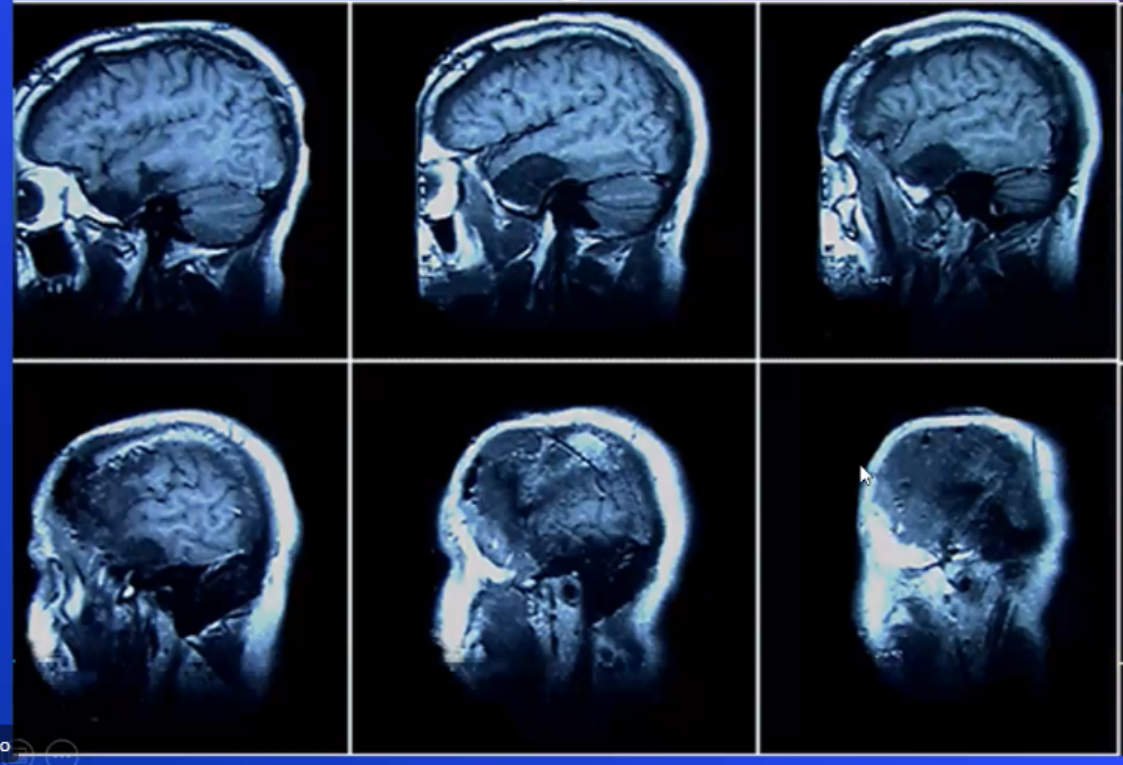
Ejemplo de imagen médica: rodilla por RM



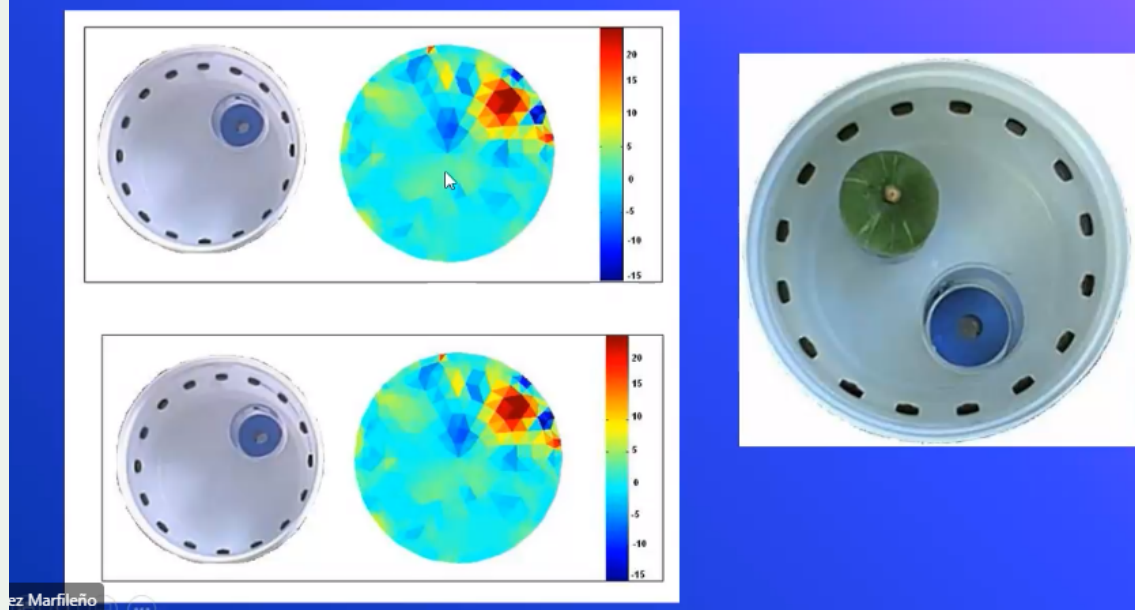
Tomografía computada



Resonancia magnética



Tomografía de impedancia eléctrica



Practica manejo de imágenes DICOM

Manejo de una imagen DICOM a distancia sobre una red local con diferentes programas de gestión DICOM.

Programación de una rutina que lee una imagen DICOM, le agrega datos patronímicos y

Generación de una imagen DICOM a partir de imágenes JPG.

▼ 22-10-2021

No hubo clase

▼ 25-10-2021

No hubo clase debido a la realización del segundo parcial

▼ Tercer Parcial

▼ 26-10-2021

Ondeletas

Funciones matemáticas que cortan los datos en diferentes componentes frecuencia

En el caso de las ondeletas, se tienen pequeños segmentos del dominio de la secuencia en el cual se concentran en un región representativa del dominio de la frecuencia

Útil para la compresión de imágenes

Se usan en procesamiento de señales, como audio, señales eléctricas, radiotransmisión e imágenes

La compresión de imágenes es deseable con gran cantidad de información

Usadas en matemáticas, Física cuántica, geología eléctrica

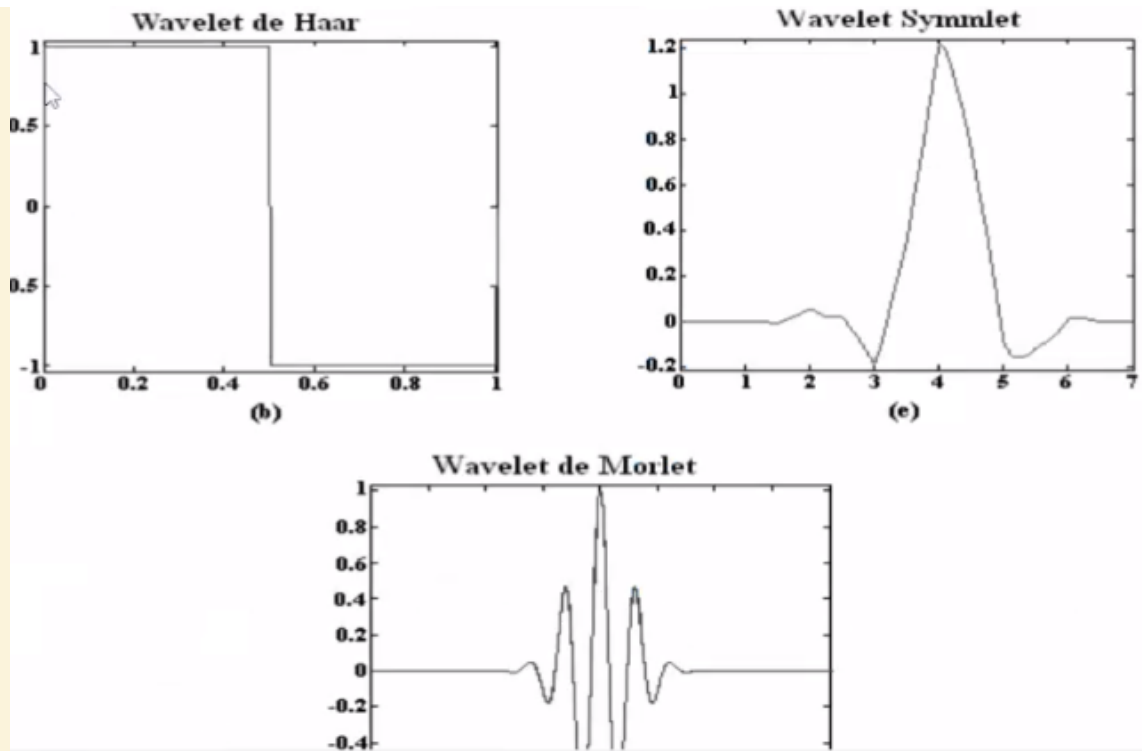
Se manejan transformadas de Fourier "windowed" o rápidas

▼ 27-10-2021

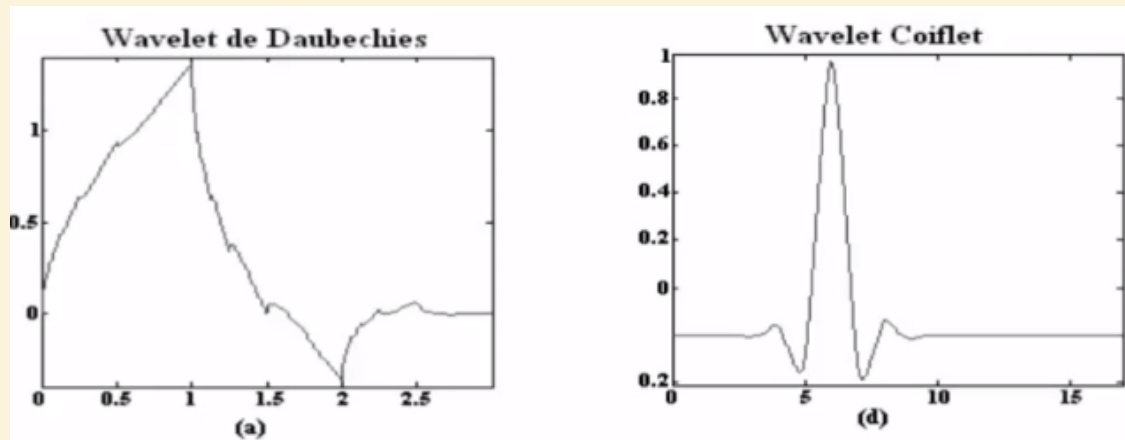
Las transformadas wavelets comprenden un amplio conjunto de formas. A lo largo del tiempo se han ido desarrollando diferentes versiones de wavelets, lo que han dado lugar a familias de wavelets. Las diferentes familias de wavelets hacen diferencias entre que tan compactas están las funciones básicas localizadas en el espacio y que tan finas son [14].

Entre las familias de wavelets más comunes se encuentran las siguientes:

- **Wavelet Haar:** El primer participante en la carrera de las wavelet fue un matemático Húngaro llamado Alfred Haar, que introdujo en 1909 las funciones que actualmente se denominan "wavelets de Haar". Estas funciones consisten simplemente en un breve impulso positivo seguido de un breve impulso negativo. Aunque los impulsos breves de las wavelets de Haar son excelentes para la enseñanza de la
- **Wavelet Haar:** El primer participante en la carrera de las wavelet fue un matemático Húngaro llamado Alfred Haar, que introdujo en 1909 las funciones que actualmente se denominan "wavelets de Haar". Estas funciones consisten simplemente en un breve impulso positivo seguido de un breve impulso negativo. Aunque los impulsos breves de las wavelets de Haar son excelentes para la enseñanza de la teoría de las wavelets, no resultan de tanta utilidad en la mayoría de aplicaciones, ya que producen líneas irregulares con picos en lugar de curvas suaves.



- **Wavelet de Morlet:** Morlet, un ingeniero de Elf-Aquitania, desarrolló su propia forma de analizar las señales sísmicas para crear componentes que estuvieran lo-



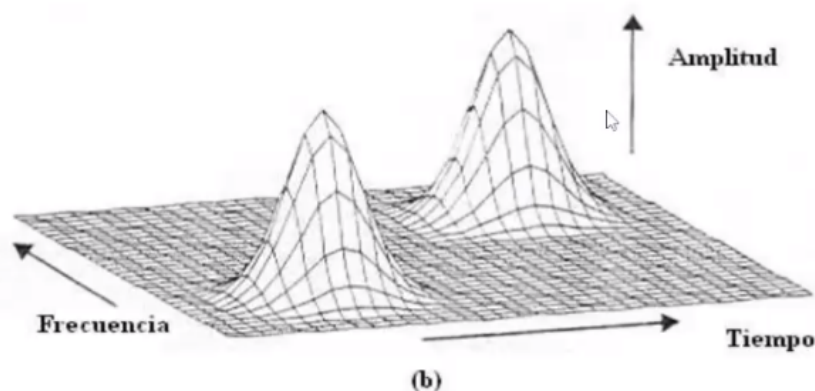


Figura 4.3: Espacio tiempo-frecuencia: (a) Señal, (b) Representación.

En un análisis imaginario, el espacio es de tres dimensiones y puede ser visto como un conjunto de imágenes. Un componente localizado puede aparecer primordialmente en el conjunto que corresponde al componente primordial de la frecuencia. La Figura 4.3 muestra una imagen que contiene dos componentes localizadas que han sido sometidas a dos filtros pasa bandas [6].

4.4. Tipos de transformadas

Así como con la transformada de Fourier, existen las tres posibilidades para las transformadas Wavelet: la transformada wavelet continua (CWT), la expansión en series wavelet y la transformada wavelet discreta (DWT). La situación es ligeramente mas complicada, sin embargo, puesto que las funciones básicas wavelet pueden o no ser ortonormales. Un conjunto de funciones básicas de wavelet pueden soportar una transformada si son o no ortonormales [6].

Para señales continuas de entrada, los parámetros de escala y tiempo pueden ser continuos, conduciendo a la transformada wavelet continua (CWT). Igualmente podría ser discreta conduciendo a la transformada wavelet discreta (DWT). En el último caso utiliza técnicas de procesamiento de señales multi taza y esta relacionado con los sistemas de codificación de sub-bandas utilizadas en compresión de imágenes y voz. Note la analogía con la transformada de Fourier (continua), series de Fourier y transformada de fourier discreta [29].

4.5. Expansión en series wavelet

Empezaremos por definir la *Expansión en series wavelet* de una función $f(x) \in L^2(\mathbf{R})$ relativa a la wavelet ψ y la función de escala φ la cual se escribe de la forma:

$$f(x) = \sum_k c_{j_0}(k) \varphi_{j_0,k}(x) + \sum_{j=j_0}^{\infty} \sum_k d_j(k) \psi_{j,k}(x) \quad (4.1)$$

donde j_0 es una escala de comienzo arbitraria y $c_{j_0}(k)$ son normalmente llamados *aproximación* o *coeficientes de escala*; la $d_j(k)$ son referidos como *detalle* o *coeficientes wavelet*.

Esto es porque la primera suma de (4.1) usan funciones de escala que proveen una aproximación a $f(x)$ en la escala j_0 (a menos que $f(x) \in V_{j_0}$ y que sea exacto). para cada escala alta $j \geq j_0$ en la segunda suma, una función de resolución fina, una suma de wavelets, se suma a la aproximación para proveer un incremento en el detalle, si las funciones de expansión forman una base ortonormal o una imagen estrecha, lo cual es siempre el caso, se calculan los coeficientes de expansión como

$$c_{j_0}(k) = \langle f(x), \varphi_{j_0,k}(x) \rangle = \int f(x) \varphi_{j_0,k}(x) dx \quad (4.2)$$

Es un conjunto de funciones básicas wavelet, $\{\psi_{s,\tau}(x)\}$, puede ser generada trasladando y escalándola wavelet básica, $\psi(x)$ y s y τ son llamados parámetros de *escala* y *traslación*, respectivamente. Dada $W_\psi(s, \tau)$ se puede obtener la *transformada continua wavelet inversa*

$$f(x) = \frac{1}{C_\psi} \int_0^\infty \int_{-\infty}^\infty W_\psi(s, \tau) \frac{\psi_{s,\tau}(x)}{s^2} d\tau ds \quad (4.6)$$

Para la cual

$$C_\psi = \int_{-\infty}^\infty \frac{|\Psi(u)|^2}{|u|} du \quad (4.7)$$

4.7. Transformada wavelet discreta

La transformada wavelet discreta es muy usada en compresión de imágenes, procesamiento y análisis. Dada un conjunto de funciones básicas ortonormales, se calcula la transformada wavelet discreta, justo como lo haríamos con cualquier otra transformada unitaria, obteniendo una función wavelet básica que sea adecuada. Como la expansión en series wavelet de la sección anterior delinea una función de una variable continua en una secuencia de coeficientes. Si la función se expande a una secuencia de números, como muestras de una función continua $f(x)$, los coeficientes resultantes son llamados *wavelet discreta*. Por lo que veremos tres técnicas para el desarrollo de la transformada wavelet discreta: (1) Teoría de bancos de filtros, (2) Multiresolución o análisis en tiempo-escala y (3) Codificación de subbanda. Para casos mas simples, las series de

4.7.1. Teoría de bancos de filtros

Investigadores en el área de análisis de voz y procesamiento de señales acústicas han usado por mucho tiempo el concepto de bancos de filtros pasa banda para descomponer la señal en componentes a diferentes frecuencias. Ciertamente, el método es un precursor del *análisis de tiempo-frecuencia*, en la cual los componentes de la señal, en la cual los componentes de la señal se muestran en espacio de dos dimensiones cuyas dimensiones son tiempo de ocurrencia y frecuencia de oscilación. Suponga que se tiene una señal compuesta de dos tonos de quiebre (sinusoides de corta duración) incrustadas en ruido aleatorio. Suponga que se analiza esta señal para detectar el número, frecuencia y posición de los tonos de quiebre. La transformada de Fourier puede reflejar el contenido entero de la señal, pero no siempre es fácil de interpretar. Por ejemplo la información

4.7.3. Codificación de subbanda

Mas allá del fondo de la transformada wavelet discreta, ahora se describe la técnica de tiempo-frecuencia llamada *codificación de subbanda*. Originalmente diseñada para codificación compacta para señales digitalizadas de audio, la codificación de subbanda pretende descomponer la señal (o imagen) dentro de los componentes de la banda estrecha (filtrado del pasa bandas) y la representa, sin redundancia, en la cual una forma posible de reconstruir la señal original sin errores. Dada la señal de banda limitada $f(t)$, que es

$$\mathfrak{F}\{f(t)\} = F(s) = 0 \quad \text{para } |s| \geq s_{\max} \quad (4.16)$$

ss muestrea la señal con un espaciado de muestra uniforme Δt para formar

$$f(i\Delta t) \quad i = 0, 1, \dots, N-1 \quad s_{\max} \leq S_N = \frac{1}{2\Delta t} \quad (4.17)$$

donde s_N es la frecuencial de Nyquist. Empieza el análisis dividiendo en partes los ejes de frecuencia en subintervalos desarticulados [6].

Aplicaciones

- *Realce de imágenes:* La DWT descompone una imagen en componentes de diferentes tamaños, posición y orientación. Como el filtrado digital en el dominio de la frecuencia de Fourier, uno puede alterar la amplitud de los coeficientes en el dominio de la transformada wavelet para obtener la transformada inversa. Esto puede selectivamente acentuar los componentes interesantes a expensas de los indeseables

- *Fusión de imágenes:* La fusión de imágenes combinan dos o más imágenes del mismo objeto en una sola imagen que es más fácil de interpretar que cualquiera de las originales. Esta técnica encuentra aplicaciones en interpretación de imágenes multispectrales, así como imágenes médicas, donde las imágenes de la misma parte del cuerpo son obtenidas por diferentes modalidades de asimilación [6].

- *Visión humana:* En los pasados 1980 David Marr empezó a trabajar en el laboratorio de inteligencia artificial del MIT en visión artificial para robots. Su teoría fue que el procesamiento de imágenes en el sistema visual humano tiene una complicada estructura jerárquica que involucra varias leyes de procesamiento. A cada nivel de procesamiento, el sistema de la retina provee una representación visual, que escala progresivamente de manera geométrica. Sus argumentos se enfocan en la detección de cambios de intensidad. Su teoría fue que los cambios de intensidad ocurren a diferentes escalas en la imagen, por tanto su detección óptima requiere el uso de operadores a diferentes escalas, este operador fue una wavelet que hoy en día es llamada "wavelet de Marr"

- *Notas musicales:* Victor Wickerhouser sugirió que un conjunto de wavelets puede ser útil en la síntesis de voz. Su idea era que un generador de paquete de wavelets simples puede reemplazar un gran número de osciladores. A través de la experimentación, un músico puede determinar las combinaciones de los paquetes de ondas que producen sonidos interesantes. Una muestra de notas producidas por un instrumento, puede ser descompuesta en sus paquetes de coeficientes wavelet. Para reproducir las notas se cargan sus coeficientes en un generador de paquetes wavelet para ver el resultado [14]. Como se ve en la Figura 4.7.

- *Compresión de imágenes:* La transformada wavelet discreta descompone una imagen en un conjunto de pequeñas imágenes sucesivamente ortonormales. Más aun, mientras el histograma de nivel de grises de la imagen original puede ser de cualquier forma, aquellos reflejos de las transformadas wavelet son comúnmente unimodales y simétricas cerca de cero. Esto simplifica el análisis estadístico de las propiedades de la señal. Como se ve en la Figura 4.8.

▼ 28-10-2021

Uso de ondeletas en aplicaciones

▼ 29-10-2021

No hubo clase

▼ 01-11-2021

Comenzamos a trabajar con lenguajes, por ahora con C

Tema 4.2b

El formato BMP

El formato **BMP**

Es uno de los formatos más simples de imágenes digitales

Fue desarrollado por Microsoft e IBM en forma conjunta, lo que explica su uso particularmente amplio en plataformas Windows y OS/2

Qué es el formato BMP

Un archivo BMP es un archivo de mapa de bits, es decir, un archivo de imagen de gráficos, con píxeles almacenados en forma de tabla de puntos que administra los colores como colores reales o usando una paleta indexada

El formato BMP ha sido estudiado de manera tal que permite obtener un mapa de bits independiente del dispositivo de visualización periférico

Cuál es la estructura de un archivo BMP

La estructura de un archivo **bitmap** consta de 3 o 4 elementos:

1. El encabezado del archivo
2. El encabezado de información del mapa de bits
3. La paleta (opcional)
4. El cuerpo de la imagen

El encabezado del archivo

El encabezado del archivo proporciona información acerca del tipo de archivo (mapa de bits) y su tamaño, así como también indica dónde comienza realmente la información de la imagen. El encabezado comprende cuatro campos:

- El **primero** corresponde a la firma (en 2 bytes), que indica que se trata de un archivo BMP con dos caracteres. **BM**, **424D** en hexadecimal, que indica que se trata de un mapa de bits de Windows. **BA** que indica que se trata de un mapa de bits OS/2. **CI** que indica que se trata de un icono de color de OS/2. **CP** indica que es un puntero de color de OS/2. **IC** indica que es un icono de OS/2. **PT** indica que es un puntero de OS/2
- El **segundo** muestra el tamaño total del archivo en bytes (codificado en 4 bytes)
- El **tercero** es un campo reservado (en 4 bytes).
- El **cuarto** indica el desajuste de la imagen (en 4 bytes), es decir, la ubicación del inicio de la información de la imagen en relación con el comienzo del archivo

Desplazamiento hexagonal	Compensación dec	Tamaño	Objetivo
00	0	2 bytes	los campo de encabezado utilizado para identificar el archivo BMP y DIB es 0x42 0x4D en hexadecimal, igual que BM en ASCII. Son posibles las siguientes entradas: BM Windows 3.1x, 95, NT, ... etc. licenciado en Letras Matriz de mapa de bits de estructura OS / 2 CI Icono de color de estructura OS / 2 CP Puntero de color const OS / 2 IC Icono de estructura OS / 2 PT Puntero OS / 2
02	2	4 bytes	El tamaño del archivo BMP en bytes
06	6	2 bytes	Reservado; el valor real depende de la aplicación que crea la imagen, si se crea manualmente puede ser 0
08	8	2 bytes	Reservado; el valor real depende de la aplicación que crea la imagen, si se crea manualmente puede ser 0
0A	10	4 bytes	El desplazamiento, es decir, la dirección de inicio, del byte donde se pueden encontrar los datos de la imagen de mapa de bits (matriz de píxeles).

El encabezado de información del mapa de bits

El encabezado de información del mapa de bits proporciona información acerca de la imagen, en especial las dimensiones y los colores

La información del mapa de bits comprende diversos campos:

- El **tamaño** del encabezado de información del mapa de bits en bytes (codificado en 4 bytes). Los siguientes valores hexadecimales son posibles según el tipo de formato BMP: 28 para Windows 3.1x, 95, NT; 0C para OS/2 1.x; F0 para OS/2 2.x.
- El ancho de la imagen (en 4 bytes), es decir, el número de píxeles contados de forma horizontal

- La **altura** de la imagen (en 4 bytes), es decir, el número de píxeles contados de forma vertical
- El número de **planos** (en 2 bytes). Este valor es siempre 1
- La **profundidad** del modelo de color (en 2 bytes), es decir, el número de bits usados para codificar el color. Este valor puede ser equivalente a 1, 4, 8, 16, 24 o 32.
- El método de **compresión** (en 4 bytes). Este valor es 0 cuando la imagen no está comprimida o 1, 2 o 3 según el tipo de compresión usado: 1 para la codificación RLE de 8 bits por píxel; 2 para la codificación RLE de 4 bits por píxel; 3 para la codificación de campo de bits, lo que significa que el color fue codificado por una máscara triple representada por la paleta

Encabezados de mapa de bits de Windows y OS / 2				
Tamaño	Nombre del encabezado	Soporte del sistema operativo	Características	Escrito por
12	BITMAPCOREHEADER OS21XBITMAPHEADER	Windows 2.0 o después OS / 2 1.x ^[10]		
64	OS22XBITMAPHEADER	OS / 2 BITMAPCOREHEADER2	Agrega medio tono. Agrega RLE y Huffman Compresión 1D.	
16	OS22XBITMAPHEADER	Esta variante del encabezado anterior contiene solo los primeros 16 bytes y se supone que los bytes restantes son valores cero. ^[9] Un ejemplo de tal caso es el gráfico pal8os2v2-16.bmp ^[10] de BMP Suite. ^[11]		
40	BITMAPINFOHEADER	Windows NT, 3.1 veces o después ^[12]	Agrega formatos de 16 bpp y 32 bpp. Agrega compresión RLE.	
52	BITMAPV2INFOHEADER	Indocumentado	Agrega máscaras de bits RGB.	Adobe Photoshop
56	BITMAPV3INFOHEADER	No documentado oficialmente, pero esta documentación fue publicada en los foros de Adobe por un empleado de Adobe con una declaración de que el estándar se incluyó en un momento en el pasado en la documentación oficial de MS. ^[12]	Agrega canal alfa máscara de bits.	Adobe Photoshop
108	BITMAPV4HEADER	Windows NT 4.0, 95 o después	Agrega tipo de espacio de color y corrección gamma	
124	BITMAPV5HEADER	Windows NT 5.0, 98 o después	Agrega Perfiles de color ICC	El GIMP

Desplazamiento (hex)	Desplazamiento (dec)	Tamaño (bytes)	OS / 2 1.x BITMAPCOREHEADER ^[11]
0E	14	4	El tamaño de este encabezado (12 bytes)
12	18	2	El ancho del mapa de bits en píxeles (16 bits sin firmar)
14	20	2	La altura del mapa de bits en píxeles (16 bits sin firmar)
16	22	2	El número de planos de color debe ser 1
18	24	2	El número de bits por píxel

Los mapas de bits de OS / 2 1.x no están comprimidos y no pueden ser de 16 o 32 bpp

La paleta de imágenes

La paleta es opcional

Cuando está definida contiene 4 bytes de forma sucesiva para cada una de las entradas, que representan el componente azul (en un byte), el componente verde (en un byte), el componente rojo (en un byte) y un campo reservado (en un byte)

La codificación de imágenes

La codificación de imágenes se realiza escribiendo en forma sucesiva los bits que corresponden a cada píxel, línea tras línea, comenzando por el píxel del extremo inferior izquierdo. Las imágenes de 2 colores usan 1 bit por píxel, lo que significa que un byte permite codificar 8 píxeles, mientras que las de 16 colores usan 4 bits por píxel, lo que significa que un byte permite codificar 2 píxeles. Por su parte, las imágenes de 256 colores usan 8 bits por píxel, lo que significa que se necesita un byte para codificar cada píxel. Las imágenes de colores reales usan 24 bits por píxel, lo que significa que se necesitan 3 bytes para codificar cada píxel, respetando la alternancia del orden de los colores para el azul, el verde y el rojo. Cada línea de la imagen debe comprender un número total de bytes que sea múltiplo de 4; si este esquema no se cumple, la línea se debe completar con todos los 0 necesarios para respetar el criterio.

Nombre del segmento de datos	Definición de estructura de Windows correspondiente	Tamaño (byte)
encabezado de archivo bmp	BITMAPFILEHEADER	14
Encabezado de información de mapa de bits	BITMAPINFOHEADER	40
Paleta		Determinado por el número de índices de color.
Datos de mapa de bits		Determinado por tamaño de imagen

La estructura del archivo de imagen de mapa de bits

ernando Gutierrez Marfileño

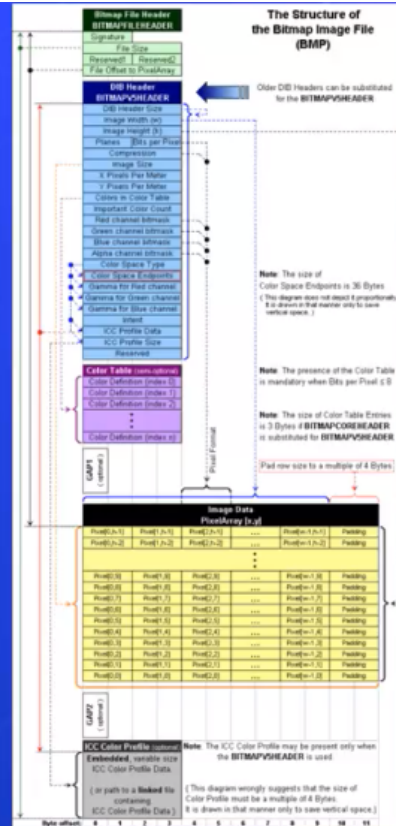


Diagrama 1

Con referencia al diagrama 1, el archivo de mapa de bits se compone de estructuras en el siguiente orden:

Nombre de la estructura	Opcional	Tamaño	Objetivo	Comentarios
Encabezado del archivo de mapa de bits	No	14 bytes	Para almacenar información general sobre el archivo de imagen de mapa de bits	No es necesario después de que el archivo se carga en la memoria
Encabezado DIB	No	Tamaño fijo (Existen 7 versiones diferentes)	Para almacenar información detallada sobre la imagen de mapa de bits y definir el formato de píxeles	Sigue inmediatamente el encabezado del archivo de mapa de bits
Máscaras de bits adicionales	si	3 o 4 DWORDs ^[6] (12 o 16 bytes)	Para definir el formato de píxeles	Presente solo en caso de que el encabezado DIB sea el BITMAPINFOHEADER y el miembro del método de compresión se establezca en BI_BITFIELDS o BI_ALPHABITFIELDS
Tabla de colores	Semi-opcional	Tamaño variable	Para definir los colores utilizados por los datos de imagen de mapa de bits (matriz de píxeles)	Obligatorio para profundidades de color ≤ 8 bits
Gap1	si	Tamaño variable	Alineación de estructura	Un artefacto de la matriz de desplazamiento de archivo a píxel en el encabezado del archivo de mapa de bits
Matriz de píxeles	No	Tamaño variable	Para definir los valores reales de los píxeles	El formato de píxel se define mediante el encabezado DIB o máscaras de bits adicionales. Cada fila de la matriz de píxeles se rellena con un tamaño múltiplo de 4 bytes
Gap2	si	Tamaño variable	Alineación de estructura	Un artefacto del campo de desplazamiento de datos del perfil ICC en el encabezado DIB
Perfil de color ICC	si	Tamaño variable	Para definir el perfil de color para la gestión del color	También puede contener una ruta a un archivo externo que contenga el perfil de color. Cuando se carga en la memoria como "DIB no empaquetado", se ubica entre la tabla de colores y Gap1. ^[7]

Los archivos BMP son imágenes ráster nativas de Windows. Los BMP también pueden almacenar varias profundidades de bit y se pueden comprimir sin pérdida de información con el método RLE (*Run Length Encoding*).

Las imágenes BMP son generalmente mucho más grandes que formatos tales como JPEG o PNG.

Este formato se puede considerar como universal dado que prácticamente todas las aplicaciones lo pueden manejar.

Este tipo de extensión se suele utilizar para imprimir imágenes, ya que poseen muy buena calidad, buen perfil de color y una imagen bidimensional más que acertada.

Las ventajas de BMP

1. La imagen parece real (la imagen es casi la misma que el objeto original)
2. Las imágenes de mapa de bits pueden capturar sombras y formas naturales
3. Las imágenes de mapa de bits se pueden transferir directamente de los archivos a la pantalla del monitor para que sea más rápido y más adecuado para mostrar en la pantalla del monitor
4. Las imágenes de mapa de bits se pueden agregar a ciertos efectos especiales para que puedan hacer que los objetos aparezcan como se desee
5. En comparación con las imágenes vectoriales, la categoría de imagen de mapa de bits es más clara y realista
6. Las imágenes de categoría de mapa de bits admiten colores de 32 bits
7. Puede producir objetos de imagen de mapa de bits a partir de objetos ambar vectoriales de una manera fácil y rápida, cuando se imprime a una resolución más baja
8. Menos uso de memoria

Desventajas de BMP

1. Cuando las imágenes de mapa de bits se reducen de tamaño, la calidad de la imagen se reducirá (la imagen resultante se vea agrietada)
2. El tamaño del archivo de imagen es mucho mayor que las imágenes vectoriales, el tamaño del archivo de mapa de bits depende de cuántos píxeles/puntos haya en la imagen
3. Cuando esta imagen cambia de tamaño (ampliación/reducción), la calidad de la imagen también cambiará
4. Cuando se imprime a una resolución más baja, los efectos obtenidos de los objetos basados en mapas de bits se verán rotos o reducidos en detalle
5. Si la imagen de visualización de mapa de bits tiene una alta resolución en la visualización en un monitor de baja resolución dará como resultado una imagen que se ve rugosa, incluso se ve borrosa en forma de cuadro (juggy) si la imagen se amplía
6. El color y el detalle de la imagen en la pantalla de mapa de bits depende de la cantidad de píxeles de color o la resolución que conforma la imagen

Características

- El formato BMP puede almacenar hasta 16,7 millones de colores en un solo archivo de imagen
- Es compatible con una resolución máxima de 32.000 por 32.000 píxeles
- Cada píxel en un archivo de mapa de bits se traduce en 1 byte de información, por lo que el número de píxeles en un archivo BMP determina el tamaño del archivo

A continuación se detalla la estructura de la cabecera de un archivo **.BMP**

Bytes	Información
0, 1	Tipo de fichero "BM"
2, 3, 4, 5	Tamaño del archivo
6, 7	Reservado
8, 9	Reservado
10, 11, 12, 13	Inicio de los datos de la imagen
14, 15, 16, 17	Tamaño de la cabecera del bitmap
18, 19, 20, 21	Anchura (píxels)
22, 23, 24, 25	Altura (píxels)
26, 27	Número de planos
28, 29	Tamaño de cada punto
30, 31, 32, 33	Compresión (0=no comprimido)
34, 35, 36, 37	Tamaño de la imagen
38, 39, 40, 41	Resolución horizontal
42, 43, 44, 45	Resolución vertical
46, 47, 48, 49	Tamaño de la tabla de color
50, 51, 52, 53	Contador de colores importantes

El Bitmap de una imagen **.BMP** comienza a leerse desde abajo a arriba, es decir: en una imagen en 24 bits los primeros 3 bytes corresponden al primer pixel inferior izquierdo.

▼ 02-11-2021

No hubo clase. Asueto

▼ 03-11-2021

Librerías para C++

- Magick++
- Boost GIL
- OpenCV
- C Image

Se decidió por C++ (objetos) con CImg

▼ 04-11-2021

No hubo clase

▼ 05-11-2021

Práctica de instalación de CImg

▼ 08-11-2021

Tema 4.3

La transformada wavelet discreta inversa

Transformada Wavelet Discreta

La idea básica de la DWT para señales de una dimensión es dividir la señal en dos partes, altas y bajas frecuencias

Las altas frecuencias representan básicamente los contornos de la señal y las bajas frecuencias se dividen nuevamente en dos partes

Este proceso continúa un número arbitrario de veces

Una de las características de la DWT es que existe la posibilidad de reconstruir la señal original a partir de los coeficientes obtenidos al aplicar la DWT

Este ultimo proceso es llamado Transformada Wavelet Discreta Inversa (IDWT)

La DWT y IDWT en dos dimensiones (m,n) pueden ser definidas como la implementación de la DWT y IDWT para una dimensión correspondientes a las dimensiones m y n separadamente

Una imagen puede ser descompuesta en una estructura en pirámide como se muestra en la figura 1

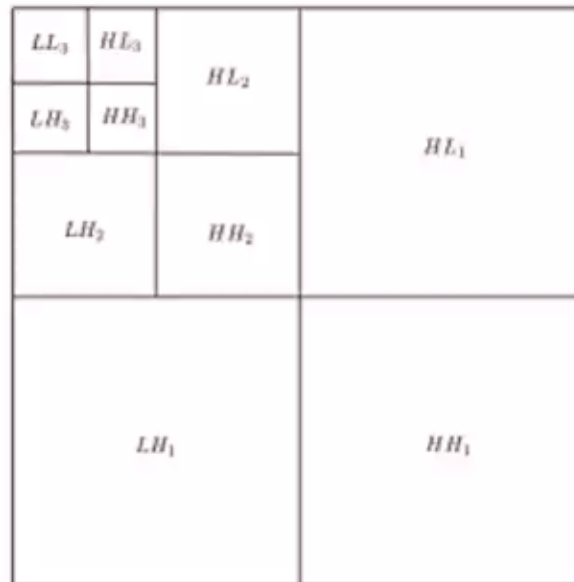


Figura 1. Pirámide al descomponer una imagen con DWT 2D.

Se puede apreciar que se obtienen cuatro tipos de coeficientes: aproximación (LL), detalles horizontales (LH), detalles verticales (HL) y detalles diagonales (HH). La aproximación contiene la mayor parte de la energía de la imagen, mientras que los detalles tienen valores próximos a cero.

En este esquema de descomposición surgen cuatro sub-bandas de un nivel de la transformada: una sub-banda de paso bajo que contiene la aproximación similar a la imagen de origen llamada sub-banda LL y tres sub-bandas de paso alto que explotan detalles de imagen en diferentes direcciones - LH y HL permite resaltar características orientadas verticalmente y horizontalmente; mientras que la wavelet de sub-banda HH tiene una orientación de composición +45 y -45.

En el siguiente nivel de la transformación, se usa la banda LL para su posterior descomposición y se descompone con sus respectivas cuatro sub-bandas.

▼ 09-11-2021

???

Tema 5.2

Compresión con pérdidas y sin pérdidas

Compresión de imagen 2

- Factores a tener en cuenta
 - Eficiencia
 - Compresión con pérdidas (lossy) -para usar-
 - O sin pérdidas (lossless) -para almacenar-
 - Algoritmos abiertos -libres- o patentados
 - A tener en cuenta tanto en la creación
 - Como en la disponibilidad a largo plazo

Compresión Run Length Encoding

El más simple. Sin pérdidas. Aplica a nivel de bits, bytes, pixels... (explicar)

Interesante en imágenes con grandes zonas planas

Hay diversas variantes y se usa en los formatos TIFF, PCX y BMP

Compresión LZ

- En honor a sus autores Abraham Lempel y Jacob Ziv en 1977-78
- El más conocido es el LZ77 que se usa en los algoritmos PKZIP y en la compresión de imagen del formato PNG
- LZ78 se usa más en compresión de imagen
- Y es la base del algoritmos LZW

Compresión Huffman (explicar)

- Desarrollado por David Huffman en 1952
- Es uno de los algoritmos más antiguos y establecidos
- Sin pérdidas
- Se usa en muchos protocolos de transmisión de datos
- Es una parte de la codificación JPEG y Deflate

Compresión Deflate

- Es un algoritmo de compresión sin pérdidas basado en LZ77 y codificación Huffman
- Desarrollado por Phil Katz en 1996
- Se usa en el algoritmo de compresión PKZIP y en la compresión de imagen del formato PNG

Compresión CCITT Grupo 3

- Grupo3.- CCITT T4 fue desarrollado en 1985 para codificar y comprimir datos de imagen de 1 bit. Su uso principal ha sido la transmisión de fax. está optimizado para documentos escaneados impresos o escritos a mano. algoritmo sin pérdidas, del cual existen dos formas: unidimensional (que es una versión modificada de la codificación de Huffman) y bidimensional, que ofrece tasas de compresión superiores. Debido a su origen como protocolo de transmisión de datos, la codificación del Grupo 3 incorpora códigos de detección de errores.

Compresión CCITT Grupo 4

- Group 4.- CCITT T.6 is a development of the two-dimensional Group 3 standard, which is faster and offers compression rates which are typically double those of Group 3.
- Like Group 3, it is lossless and designed for 1-bit images.
- However, being designed as a storage rather than transmission format, it does not incorporate the error detection and correction functions of Group 3

o Gutierrez Marfileño

Compresión LZW

- by Terry Welch in 1984, as a modification of the LZ78 compressor.
- Lossless
- can be applied to almost any type of data.
- most commonly used for image compression.
- effective on images with colour depths from 1-bit to 24-bit
- The patent for the LZW algorithm is owned by Unisys Corporation, which has licensed its use in a variety of file formats, most notably CompuServe's GIF format
- It should be noted that the licensing applies to implementations of the LZW algorithm, and not to individual files which utilise it.
- The US and the UK patent expired in 2004.
- LZW compression is encountered in a range of common graphics file formats, including TIFF and GIF.

Compresión: JPEG

- 1990. Colour and greyscale images.
- JPEG is a lossy technique.
- Best compression rates with complex 24-bit (True Colour) images.
- It discards image data which is imperceptible to the human eye, using a technique called Discrete Cosine Transform (DCT).
- It then applies Huffman encoding to achieve further compression.
- JPEG comprises a baseline specification + optional extensions, including:
 - Progressive JPEG. Useful for applications which need to stream image data.
 - JPEG always involves some degree of lossy compression.
 - Repeated saving of an image lead to increasing degradation of the quality.

Compresión JPEG2000

- JPEG 2000 is a replacement for the JPEG algorithm
 - lossy and lossless compression
 - **wavelet compression**
 - higher compression rates
 - with a lower corresponding reduction in image quality.
- JPEG 2000 may utilise some patented technologies.
- But intended to be made available license- and royalty-free.
- Minimum file interchange format (JP2),
 - in a similar manner to JFIF and SPIFF.
- Support for JPEG 2000 is now beginning to appear in a number of commercial software packages.

Compresión PNG

- PNG compression was developed in 1996
 - as part of the PNG file format.
- PNG compression uses the Deflate compression method.
- It is a lossless algorithm and is effective with colour depths from 1-bit to 48-bit.
- Unencumbered by patent and free to use.
- It is implemented only in the PNG file format.
- It is a W3C recommendation.
- Version 1.2 is intended to be released as an ISO standard

Compresión fractal 1

- Fractal compression uses the mathematical principles of fractal geometry to identify redundant repeating patterns within images.
- These matching patterns may be identified through performing geometrical transformations, such as scaling and rotating, on elements of the image.
- Once identified, a repeating pattern need only be stored once, together with the information on its locations within the image and the required transformations in each case.
- Fractal compression is extremely computationally intensive, although decompression is much faster.

Compresión fractal 2

- It is a lossy technique, which can achieve large compression rates.
- Unlike other lossy methods, higher compression does not result in pixelation of the image and, although information is still lost, this tends to be less noticeable.
- Fractal compression works best with complex images and high colour depths.
- The original fractal compression algorithm was developed by Michael Barnsley in 1991.
- However, the algorithm is patented and supported by few commercial products.
- It is not implemented in any common graphics file formats.

Conclusión

- **Algorithm Lossiness Efficient with**
- RLE Lossless Monochrome or images with large blocks of colour
- LZ Compressors Lossless All images
- Huffman Encoding Lossless All images
- Deflate Lossless All images
- CCITT Group 3 & 4 Lossless Monochrome images
- LZW Lossless All images
- JPEG Lossy (lossless extension available) Complex, True Colour images
- JPEG 2000 Lossy, lossless supported Complex, True Colour images
- PNG Lossless All images
- Fractal Lossy Complex, True Colour images

Algorithm	Lossiness	Efficient with
RLE		Lossless
LZ Compressors	Lossless	All images
Huffman	Encoding	Lossless
Deflate	Lossless	All images
CCITT Group3&4	Lossless	Monochrome images
LZW	Lossless	All images
JPEG	Lossy (lossless extension)	Complex, True Colour images
JPEG 2000	Lossy, lossless supported	Complex, True Colour images
PNG	Lossless	All images
Fractal	Lossy	Complex, True Colour images

▼ 11-11-2021

No hubo clase

▼ 12-11-2021

No hubo clase

▼ 15-11-2021

No hubo clase. Asueto

!!! NO HE PASADO APUNTES DESDE AQUÍ !!!

▼ 16-11-2021

Código Huffman

- Es un método usado para compresión de datos. El término se refiere al uso de una tabla de códigos de longitud variable para codificar un determinado símbolo (como puede ser un carácter en un archivo), donde la tabla ha sido rellena de una manera específica basándose en la probabilidad estimada de aparición de cada posible valor de dicho símbolo.

El proceso de construcción del árbol consiste en ordenar los caracteres únicos por frecuencia relativa, de menor a mayor y de izquierda a derecha. Ponemos cada uno de estos y sus frecuencias relativas en nodos conectados por ramas. Posteriormente se forma un nodo intermedio que agrupa a los dos nodos hoja que tienen menor peso (frecuencia de aparición). El nuevo nodo intermedio tendrá como nodos hijos a esos dos nodos hoja y su campo peso será igual a la suma de los pesos de los nodos hijos. Los dos nodos hijos se eliminan de la lista de nodos, sustituyéndolos por el nuevo nodo intermedio. El proceso se repite hasta que sólo quede un nodo en la lista. Éste último nodo se convierte en el nodo raíz del árbol de Huffman.

Funcionamiento:

- Los símbolos de la fuente se listan en orden de probabilidades decrecientes. A los dos símbolos de probabilidad más baja (penúltimo y último) se le asignan un 0 y un 1, respectivamente.
- Estos dos símbolos se consideran como si se combinaran en un nuevo símbolo de fuente con probabilidad igual a la suma de las dos probabilidades originales. La probabilidad obtenida se pone en la lista de acuerdo con su valor.
- El procedimiento se repite hasta que nos quedamos con la lista final de los únicos símbolos, a los cuales se les asignarán un 0 y un 1.
- El código para cada símbolo se determina procediendo hacia atrás y siguiendo la frecuencia de ceros y unos asignados a cada símbolo.

Básicamente, la idea de Huffman fue identificar los píxeles que aparecen con mayor frecuencia en una imagen y asignarles representaciones cortas. Los píxeles que ocurren con menor frecuencia en una imagen se les asigna representaciones largas. Se puede entender mejor esta idea analizando un ejemplo de codificación de Huffman:

▼ 17-11-2021

La codificación Huffman tiene el inconveniente de asignar una palabra de código con un número entero de bits para cada símbolo ya que el número de bits óptimo para un símbolo, depende del contenido de la información y normalmente es un número fraccionario

Por este motivo, la eficiencia en la compresión de los códigos de longitud variable es particularmente pobre para símbolos con probabilidades mayores de 0'5

Sin embargo, la codificación aritmética asigna un código binario (normalmente largo) a todo un mensaje, resultando un número fraccionario de bits por símbolo

El método lee todos los símbolos de la cadena que representa al mensaje y calcula la probabilidad de cada símbolo

Con dichas probabilidades va estrechando consecutivamente un cierto intervalo inicial de valores reales

Especificar un intervalo más estrecho requiere más bits, por lo que los intervalos construidos por el algoritmo requieren cada vez más y más números para especificar sus límites

El algoritmo está diseñado de forma que un símbolo con probabilidad alta estreche el intervalo menos que un símbolo con baja probabilidad, con lo que resulta que los símbolos con alta probabilidad contribuyen con pocos bits a la salida, consiguiendo así la compresión

El rango de valores reales inicial es $[0,1)$ (la notación $[a, b)$ significa el rango de números reales desde a hasta b , incluyendo a pero no incluyendo b) y la salida del codificador aritmético se interpreta como un número dentro de ese rango

Sólo se tiene en cuenta la parte decimal de dicho código, por lo que el código 9746509 se interpreta como 0.9746509

Procedimiento

Para entender cómo se divide el intervalo $[0,1)$ en subintervalos, a continuación se representa un ejemplo de codificación aritmética del mismo mensaje que se codificó en el ejemplo de codificación Huffman

Al igual que en el ejemplo anterior, el alfabeto fuente está formado por seis símbolos cuyas probabilidades de ocurrencia se recogen en la tabla 2.1 junto con los subintervalos de $[0,1)$ que se asignan a cada símbolo dependiendo de su probabilidad de ocurrencia

El mensaje a codificar es la serie de símbolos $b_1b_2b_3b_4b_5b_6$

Símbolos Fuente	Probabilidad de Ocurrencia	Subintervalos Asociados
s_1	0.3	$[0, 0.3)$
s_2	0.1	$[0.3, 0.4)$
s_3	0.2	$[0.4, 0.6)$
s_4	0.05	$[0.6, 0.65)$
s_5	0.1	$[0.65, 0.75)$
s_6	0.25	$[0.75, 1)$

Tabla 2.1 Ejemplo de codificación aritmética: probabilidades de los símbolos.

Inicialmente se parte del intervalo $[0,1)$, y este intervalo se va reduciendo según los subintervalos asociados a los símbolos que forman la cadena a codificar

Al codificar un nuevo símbolo, la longitud del intervalo inicial se reduce de forma proporcional al valor de la longitud del subintervalo asociado a dicho símbolo del siguiente modo:

$$Long_{nueva} = Long_{actual} \cdot p(b_i)$$

donde $p(b_i)$ es la probabilidad de ocurrencia del símbolo

Inicialmente se parte del intervalo $[0,1)$, y este intervalo se va reduciendo según los subintervalos asociados a los símbolos que forman la cadena a codificar

Al codificar un nuevo símbolo, la longitud del intervalo inicial se reduce de forma proporcional al valor de la longitud del subintervalo asociado a dicho símbolo del siguiente modo:

$$Long_{nueva} = Long_{actual} \cdot p(bi)$$

donde $p(bi)$ es la probabilidad de ocurrencia del símbolo

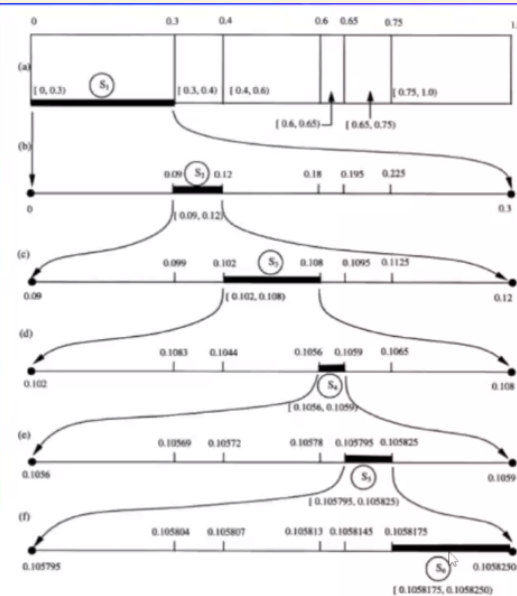
El límite inferior del intervalo resultante después de codificar un símbolo, se calcula mediante la siguiente expresión:

$$Lbm_bnf_{nuevo} = Lbm_bnf_{actual} + Lbm_bnf_{símbolo} \cdot Long_{actual}$$

Y finalmente el límite superior del nuevo intervalo se calcula mediante la suma del nuevo límite inferior más la nueva longitud calculada:

$$Lbm_bup_{nuevo} = Lbm_bnf_{nuevo} + Long_{nuevo}$$

En la siguiente figura se muestra gráficamente todo el proceso:



Codificación de video de alta eficiencia y de contenido de pantalla (HEVC-SCC), es un estándar de compresión de video diseñado como parte del MPEG-H proyecto como sucesor del ampliamente utilizado Codificación de video avanzada (AVC). En comparación con AVC, HEVC ofrece de un 25% a un 50% mejor compresión de datos al mismo nivel de calidad de video, o una calidad de video sustancialmente mejorada a la misma tasa de bits. Admite resoluciones de hasta 8192×4320 , incluidas 8K UHD. La diferencia del AVC principalmente de 8 bits, el perfil Main10 de mayor fidelidad de HEVC se ha incorporado a casi todo el hardware de soporte.

Las extensiones de Codificación de contenido de pantalla (**SCC**) mejorarán la capacidad de compresión para videos que contienen una porción significativa de gráficos, texto o animación renderizados (en movimiento o estáticos) en lugar de (o además) escenas de video capturadas por la cámara. Las aplicaciones incluyen pantallas inalámbricas, acceso remoto al escritorio de la computadora y uso compartido de la pantalla en tiempo real para videoconferencias.

El diseño de la mayoría de los estándares de codificación de video tiene como objetivo principal tener la mayor eficiencia de codificación

La eficiencia de codificación es la capacidad de codificar video a la tasa de bits más baja posible mientras se mantiene un cierto nivel de calidad de video

Hay dos formas estándar de medir la eficiencia de codificación de un estándar de codificación de video, que son usar una métrica objetiva, como relación pico señal-ruido (PSNR), o para utilizar la evaluación subjetiva de la calidad del video

El diseño de la mayoría de los estándares de codificación se considera que la evaluación subjetiva de la calidad del video es la forma más importante de medir un estándar de codificación de video, ya que los humanos percibimos la calidad del video subjetivamente

HEVC está dirigido a pantallas HDTV de próxima generación y sistemas de captura de contenido que presentan tasas de cuadros escaneados progresivos y resoluciones de pantalla de QVGA (320x240) a 4320p (7680x4320), así como calidad de imagen mejorada en términos de nivel de ruido, espacios de color y dinámica distancia

HEVC fue diseñado para mejorar sustancialmente la eficiencia de codificación en comparación con AVC, es decir, para reducir los requisitos de tasa de bits a la mitad con una calidad de imagen comparable, a expensas de una mayor complejidad computacional

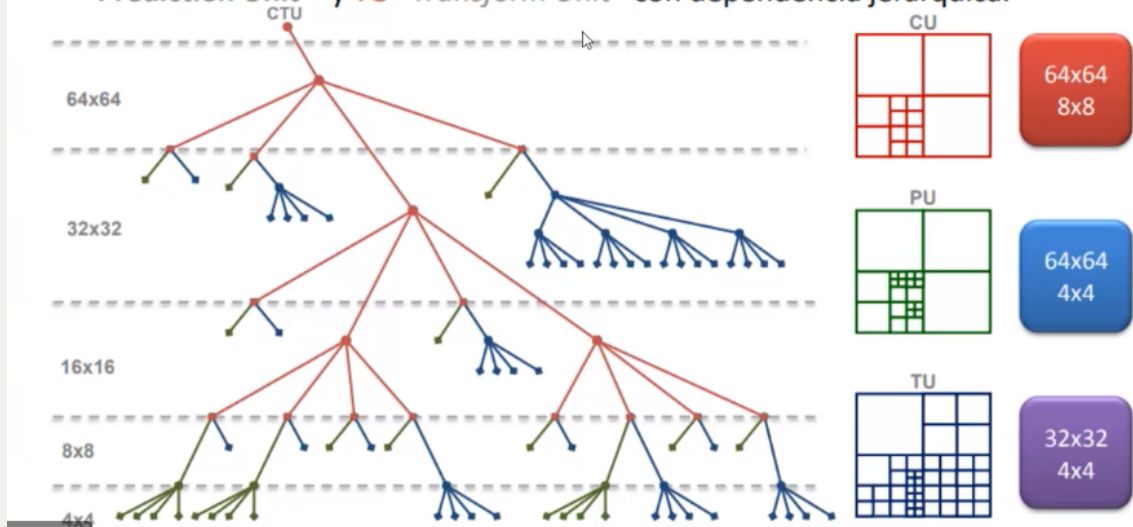
HEVC fue diseñado con el objetivo de permitir que el contenido de video tenga una relación de compresión de datos de hasta 1000: 1

Dependiendo de los requisitos de la aplicación, los codificadores HEVC pueden compensar la complejidad computacional, la tasa de compresión, la solidez a errores y el tiempo de retardo de codificación

Dos de las características clave en las que se mejoró HEVC en comparación con AVC fue la compatibilidad con video de mayor resolución y métodos mejorados de procesamiento paralelo

HEVC está dirigido a pantallas HDTV de próxima generación y sistemas de captura de contenido que cuentan con velocidades de cuadro de escaneo progresivo y resoluciones de pantalla de QVGA (320×240) a 4320p (7680×4320), así como una calidad de imagen mejorada en términos de nivel de ruido, color espacios y rango dinámico

- Además del CTU, HEVC define 3 nuevas unidades **CU** "Coding Unit", **PU** "Prediction Unit" y **TU** "Transform Unit" con dependencia jerárquica.



Definiciones

Coding Tree Block (CTB):

La IMAGEN se divide en estructuras cuadradas $N \times N$ llamadas coding tree blocks (CTBs). El tamaño de N de los CTB es escogida por el codificador (16x16, 32x32, 64x64). La luminancia cubre un área cuadrada de muestras de tamaño $N \times N$, mientras que la crominancia cubre en cada componente un área cada una de ellas $N/2 \times N/2$ (en 4:2:0).

Coding Block (CB):

Cada CTB es la raíz de árbol de codificación que se utiliza para dividir el CTB en coding blocks (CBs). El tamaño del CB puede ir desde el mismo tamaño del CTB hasta un tamaño mínimo de 8x8. El tamaño adaptativo es debido a que los CBs representan las hojas del Quadtree que forma el CTB.

Cada Coding Block es la raíz para dos nuevos árboles, uno de predicción (Prediction tree) y otro de transformada (Transform tree).

Pediction Block (PB):

Cada una de las hojas del pediction tree es lo que se considera un Prediction Block, si bien este árbol sólo tiene un nivel y realmente describe cómo un CB puede ser particionado entre diferentes modos de particionado en los inter-PB. El prediction tree especifica la posición y tamaño de los PB.



▼ 19-11-2021

No hubo clase

▼ **22-11-2021**

No hubo clase

▼ **23-11-2021**

Aclaración de criterios de evaluación de la materia para el final

▼ **24-11-2021**

▼ **25-11-2021**

▼ **26-11-2021**