



Sistemas de Visión Artificial

Tema 1. Fundamentos de Visión Artificial

Autores: Sira Palazuelos, Luis M. Bergasa , Manuel Mazo,
M. Ángel García, Marisol Escudero, J. Manuel Miguel
Departamento de Electrónica. Universidad de Alcalá.

Índice

1. Aspectos generales sobre visión por computador
2. Sistemas de Iluminación
3. Óptica
4. Cámaras
5. Adquisición y geometría de formación de imágenes
6. Radiometría
7. Fotometría
8. Digitalización
9. Geometría de la cámara
10. Introducción al color



1. Aspectos generales sobre visión por computador

¿Qué es la Visión por Computador?

Visión por Computador

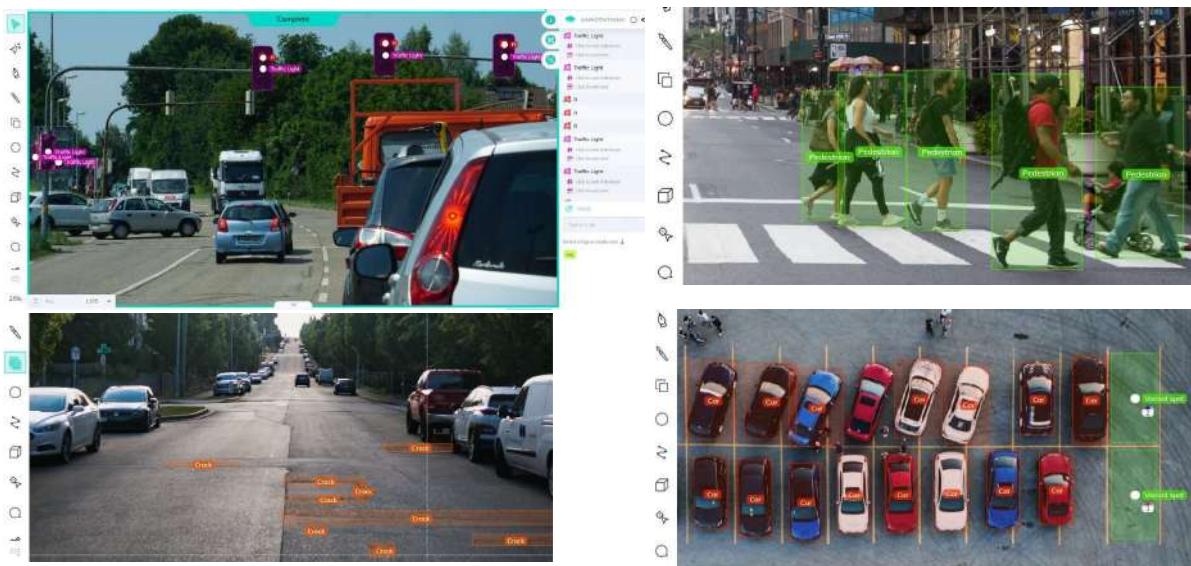
- Es la ciencia que estudia los procesos de adquisición, procesado, análisis e interpretación de la información procedentes de imágenes 2D de un mundo 3D.



- Escena de tráfico.
- Número de vehículos.
- Tipo de vehículos.
- Localización de obstáculos próximos.
- Valoración de la densidad (congestión).

<https://www.v7labs.com/blog/computer-vision-applications>

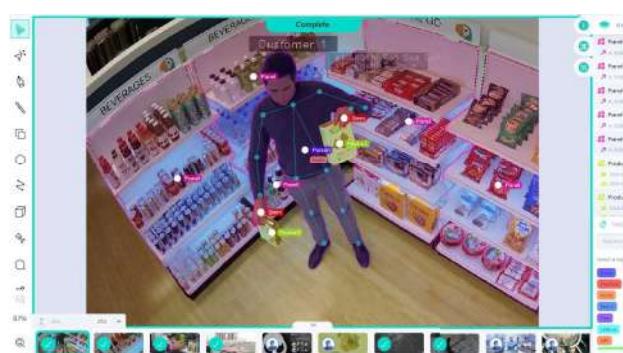
- **Robótica (Guiado y Navegación)**
- **Sistemas de transporte:** coche autónomos (detección de semáforos, de peatones, análisis de flujo de tráfico, inspección del estado de las carreteras, cansancio de los conductores, dirección de la mirada...)



<https://www.v7labs.com/blog/computer-vision-applications>

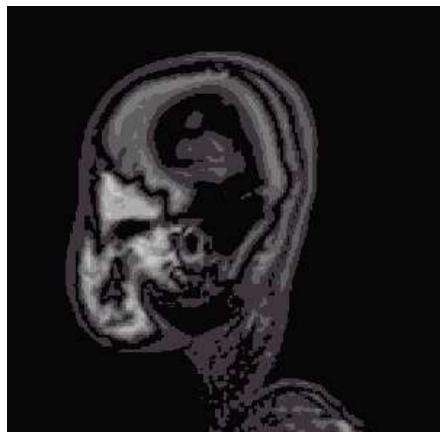
□ Aplicaciones de seguridad

- Estimación de pose, tráfico y conteo de personas, detección de comportamientos anómalos, detección de EPIs.
- <https://www.youtube.com/watch?v=HHXRqCGCRCs>



<https://www.v7labs.com/blog/computer-vision-applications>

- **Medicina:** tratamiento de imágenes médicas, detección de cáncer en imágenes de piel.

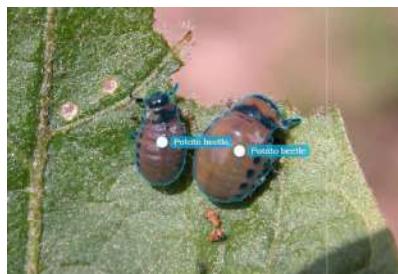


Más usos, imágenes y vídeos en:

<https://www.v7labs.com/blog/computer-vision-applications>



- Reconstrucción 3D, objetos ocultos
- Usos industriales: Inspección, control de calidad, mantenimiento predictivo
- Identificación
- Agricultura
- Deporte
- ...



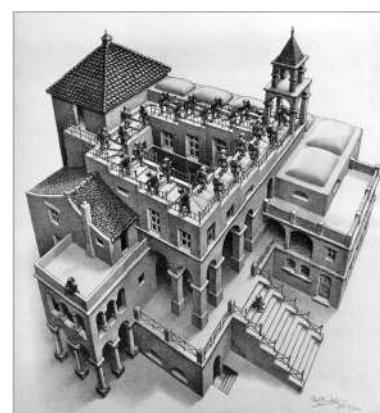
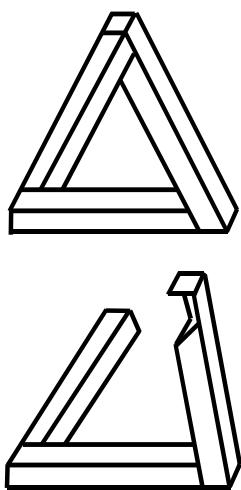
<https://www.v7labs.com/blog/computer-vision-applications>

- Las imágenes son ambiguas: proyección 3D a 2D.
- Variaciones naturales en las clases de objetos:
 - Color, textura, tamaño, formas, partes, relación.
- Variaciones en el procesamiento de imágenes:
 - Iluminación (brillos, sombras, contraste),
 - Distorsión de proyección, punto de vista, occlusiones;
 - Ruido, características de sensor y óptica.
- Varios aspectos interrelacionados: localización, reconocimiento, etc.
- Gran cantidad de información a tratar (gran potencia de cálculo).



Blueberry muffin or Chihuahua
Pinterest

- Problema de representar en 2D objetos 3D

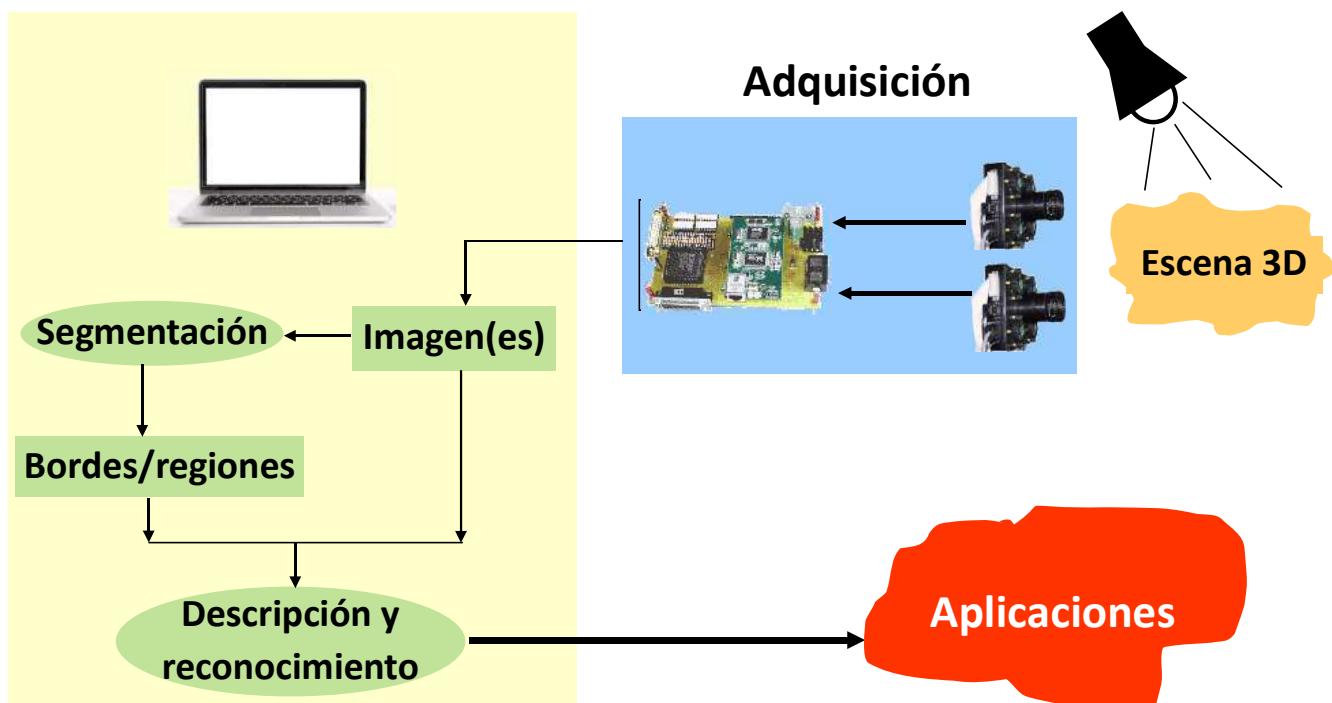


Ascending and descending
[M.C. Escher, 1960]

[https://www.archdaily.co/co/02-327413/monument-valley-un-juego-de-
arquitectura-e-ilusiones-opticas/52d932fae8e44efb240000c7?next_project=no](https://www.archdaily.co/co/02-327413/monument-valley-un-juego-de-arquitectura-e-ilusiones-opticas/52d932fae8e44efb240000c7?next_project=no)

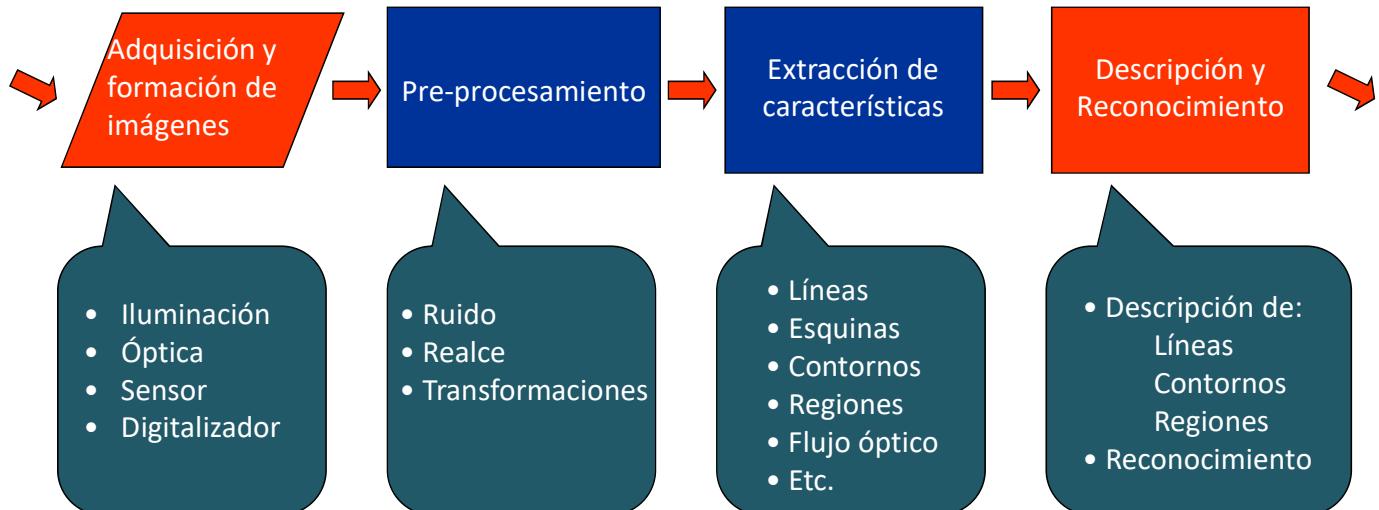
- El **sistema de visión humano** permite reconocer objetos, incluso complejos, en fracciones de segundo.
- La **retina** de un ojo contiene, aproximadamente, 10^8 células de visión especializadas y 4 capas de neuronas, todas capaces de llevar a cabo unos 10^7 operaciones por segundo.
- Cuando la imagen alcanza el **cerebro** (10^{11} neuronas y 10^{14} interconexiones) se efectúa un tratamiento altamente complejo.
- Se estima que el 60% de la **corteza cerebral** se dedica a tratar información visual.
- Los sistema de visión artificial tienen un largo y difícil camino por delante, antes de llegar a resultados comparables a los de la visión humana, aunque hay algunas tareas en las que ya han superado a los humanos (identificación de caras).

Diagrama de bloques de un sistema de visión artificial



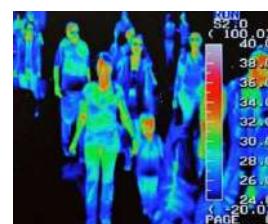
Escena 3D

Aplicación



Formación y tipos de imágenes

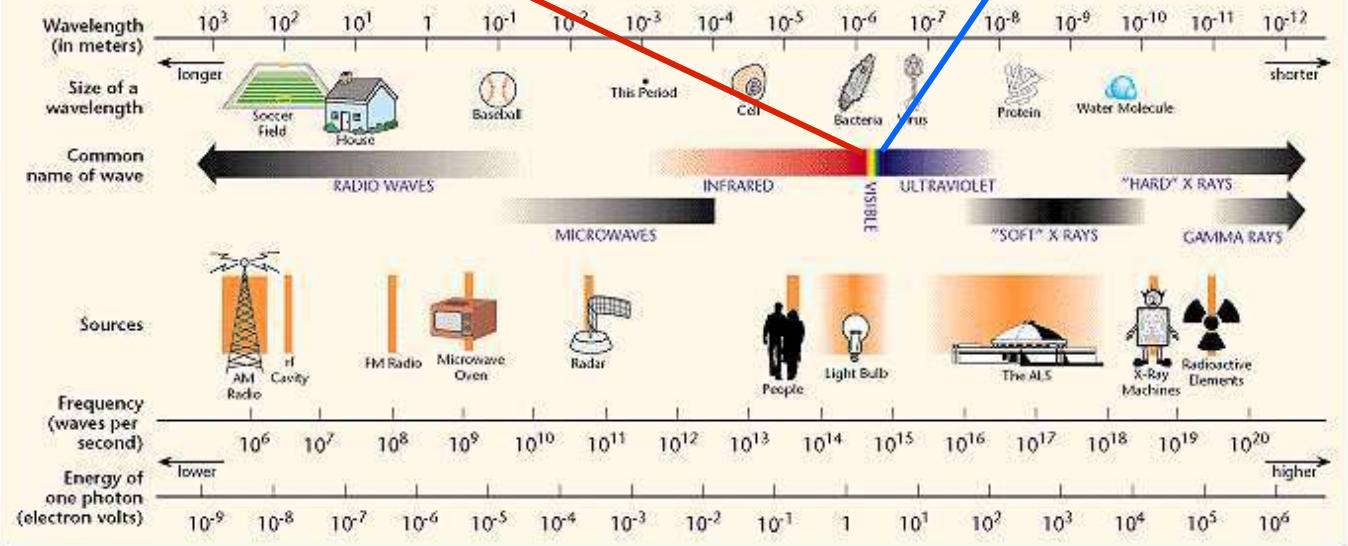
- **Formación de imágenes:** las imágenes se forman cuando un **sensor** registra la **radiación** que interactúa con los **objetos físicos** del mundo 3D.
- **Tipos de imágenes (sensores):**
 - **Fotografía** (blanco/negro y color). Luz reflejada.
 - **Imágenes de distancia:** distancia.
 - **Tomografía:** densidad de tejido.
 - **Infrarrojo:** se suele referir al IR cercano.
 - **Térmicas:** IR lejano, captan calor.
 - **Hiperespectrales:** captan más longitudes de onda → imagen más precisa (captan vegetación, agua, minerales,...)
 - Etc..
- Nosotros nos centraremos en los sistemas de **luz reflejada**, aunque el procesamiento de los demás es similar.
- <https://www.luismonje.com/portfolio/>



Espectro visible



THE ELECTROMAGNETIC SPECTRUM





2. Sistemas de Iluminación

Iluminación - Introducción

- Un correcto **sistema de iluminación** contribuye de forma notable al éxito de los sistemas de visión (sobre todo en aplicaciones industriales).
- Las características de las fuentes luminosas tienen una gran repercusión sobre las **prestaciones** y el **coste** del sistema de visión (SV).
- Un buen sistema de iluminación puede **reducir la complejidad** de los algoritmos de procesamiento. Hay que considerar la **Intensidad lumínica, duración, flexibilidad de diseños y precio**.
 - **Las fuentes de luz más típicas en los SV son:**
 - Fluorescentes
 - LED's
 - Láser
 - Gas estroboscópico (Xenon)
 - **Las técnicas de iluminación más frecuentes en los SV son:**
 - Directa
 - Lateral
 - Estroboscópica
 - Por campo oscuro (dark field)
 - Difusa coaxial
 - Por contraste (backlight)
 - Difusa tipo domo
- <https://infaimon.com/enciclopedia-de-la-vision/>

□ Fluorescente

- Tubo de vidrio que contiene fósforo y otros elementos que emite luz al recibir radiación UV de onda corta
- La principal ventaja es su **eficiencia** energética
- Su uso en aplicaciones industriales está limitado por escasa variedad de formas
- Se utilizan fluorescentes especiales que funcionan a $f > 25$ kHz, si no, generan efecto parpadeo
- La principal ventaja frente a los LED:
 - Sólo se cambia el fluorescente
- Vida media es de unas 10.000 h



Paneles



Circulares



Tubos de alta
y Ultra alta
frecuencia

□ LED (Light Emitting Diodes)

- Intensidad de la iluminación continua
- Proporcionan gran intensidad de iluminación
- Larga vida, aproximadamente 100.000 horas
- Emiten luz en color rojo, verde, azul blanco... IR
- Tecnología de LED chip-on-board (LED COB)
 - Permite una gran concentración de LED en una superficie muy reducida
 - Aumentar el poder lumínico



Rectangulares

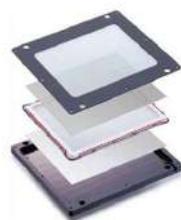


Anillos



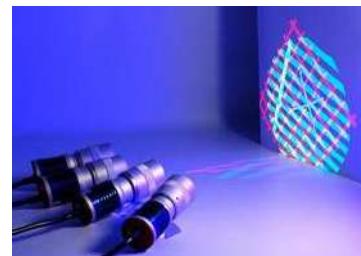
Paneles

- ## □ Light conducting surfaces.
- Basadas en tecnología LED: se proyecta la luz desde uno de los lados, y es guiada por microsurcos, creando una iluminación homogénea para cualquier superficie diseñada.



□ Láser

- Luz estructurada.
- Diferentes patrones: puntos, líneas rectas, circunferencias, cruces, cuadrados, matrices de puntos y múltiples líneas.
- Simplifican mucho la imagen.
- Aplicaciones:
 - Para marcar el trazo a seguir.
 - Para determinar la tercera dimensión de los objetos.
 - Realizar medidas.



□ Fibra óptica

- Funcionamiento: Un haz de fibras ópticas conduce la luz de una fuente hasta un adaptador
 - Fuentes LED, halógenas o de xenón
 - Haces de fibra óptica de una longitud < 5 m
 - Adaptadores de forma circular, lineal, puntual o de panel...
- Proporciona la luz más intensa de todos los tipos de iluminación que se utilizan en visión artificial
- Proporciona luz fría, ideal para aplicaciones:
 - Donde no se pueda emitir calor.
 - En entornos deflagrantes.
- Vida media entre 1000 y 2000 h



Fuente Fibra Óptica

Terminadores de iluminadores



□ Iluminación directa

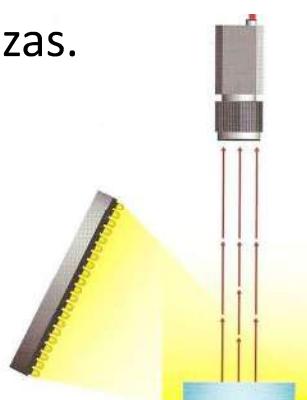
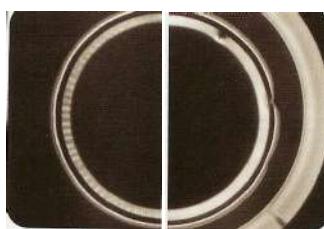
- Es la técnica más común.
- La cámara recibe la luz reflejada del objeto.
- Esta luz se puede conseguir con:
 - Iluminadores puntuales.
 - Iluminadores de áreas.
 - Iluminadores lineales.
- Es especialmente útil en superficies con pocos reflejos:
 - Papel,
 - Tela...



<https://infaimon.com/category/enciclopedia-de-la-vision/iluminacion/>

□ Iluminación lateral

- La cámara recibe la luz reflejada del objeto.
- Se ilumina de forma lateral a la posición de la cámara.
- Se utiliza para resaltar detalles de piezas.

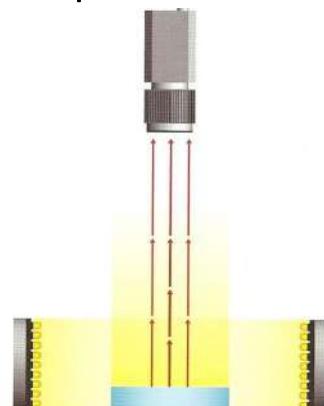


□ Iluminación estroboscópica

- Es luz pulsada.
- Se utiliza con piezas que se mueven a gran velocidad.
- <https://www.youtube.com/watch?v=tWHES1ibEZY>
- <https://www.youtube.com/watch?v=8a2Vw0c1hJ8>

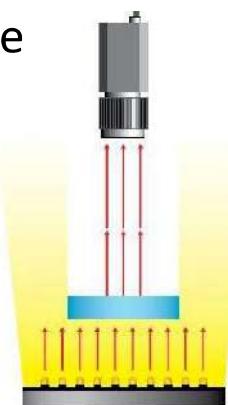
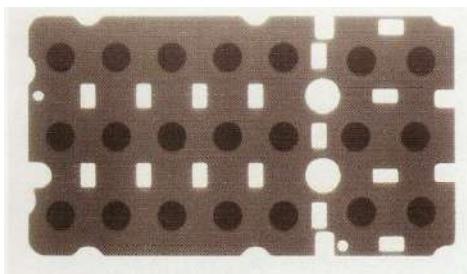
□ Iluminación por campo oscuro (dark field)

- Esta técnica resalta:
 - Defectos superficiales.
 - Grietas.
 - Surcos.
- Se utiliza para leer códigos grabados en una superficie.
- Se utilizan anillos de luz
 - Iluminan en dirección perpendicular a la cámara



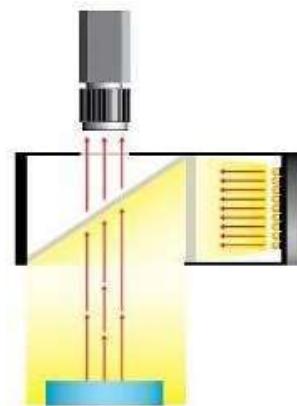
□ Iluminación por contraste (backlight)

- Se sitúa al objeto entre la iluminación y la cámara.
- Medidas muy precisas y reconocimiento de la silueta por contraste (contraluz).
- No permite ver detalles superficiales de piezas.
- En materiales translúcidos o transparentes se pueden medir:
 - Manchas,
 - Grietas,
 - Rayas.



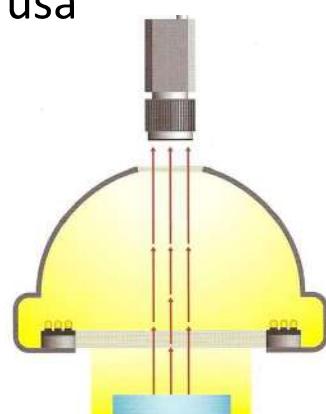
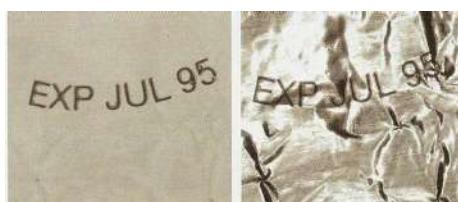
□ Iluminación difusa coaxial

- Iluminación sobre el mismo eje utilizando un espejo semitransparente (*beam splitter*).
- Se utiliza para iluminar objetos muy reflectantes:
 - Etiquetas reflectantes
 - PCB
 - Aluminio



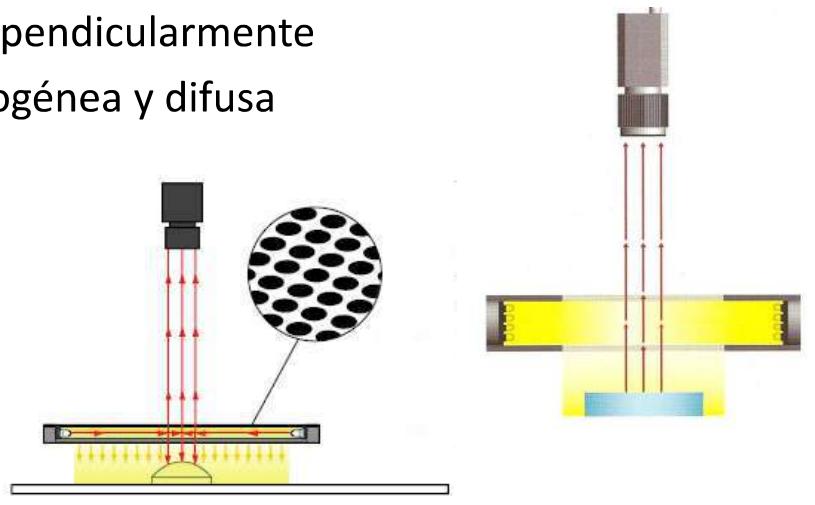
□ Iluminación difusa tipo domo

- Iluminación de día nublado (no produce sombra).
- Máximo rendimiento en iluminación difusa
- Combina iluminación:
 - Reflectante (cúpula esférica)
 - Coaxial
- Se utiliza para iluminar superficies especulares complejas
 - Espejos
 - Latas
 - CDs...



□ Iluminación difusa tipo domo plano

- Espesor muy reducido (< 1 cm, al contrario que el domo que requiere gran espacio entre objeto y cámara)
- Lámina de material especial transparente
 - Difracta la luz perpendicularmente
 - Iluminación homogénea y difusa



Herramienta on-line de selección del iluminador

Asesor de iluminación de COGNEX

www.cognex.com → Recursos →
Herramientas interactivas →
Asesor de iluminación

Si el menú es inglés:

Home → Resources →
Interactive Tools →
Lighting Advisor

<https://www.cognex.com/es-es/resources/interactive-tools/lighting-advisor>

Permite probar sobre objetos de distintas características (brillo, mate, etc.) diferentes tipos de iluminación, configurando parámetros como polarización, distancia al objeto, etc.



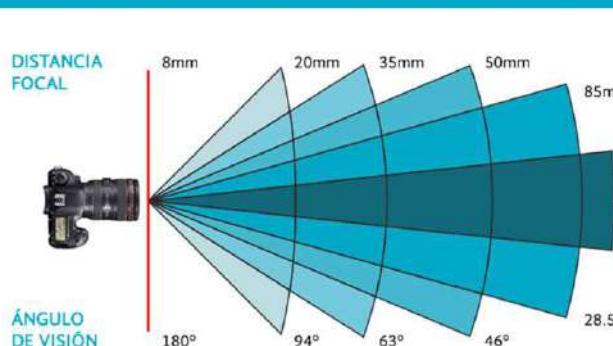
Actividad: probar distintas configuraciones y llenar el siguiente cuestionario:

<https://forms.office.com/e/zkaE1nFAxb>



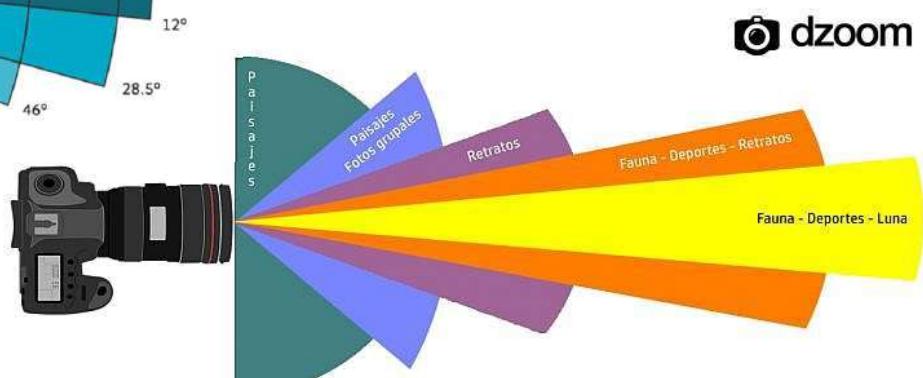
3. Óptica

Óptica – Ángulo de Visión



[https://blog.foto24.com/
distancia-focal-objetivo/](https://blog.foto24.com/distancia-focal-objetivo/)



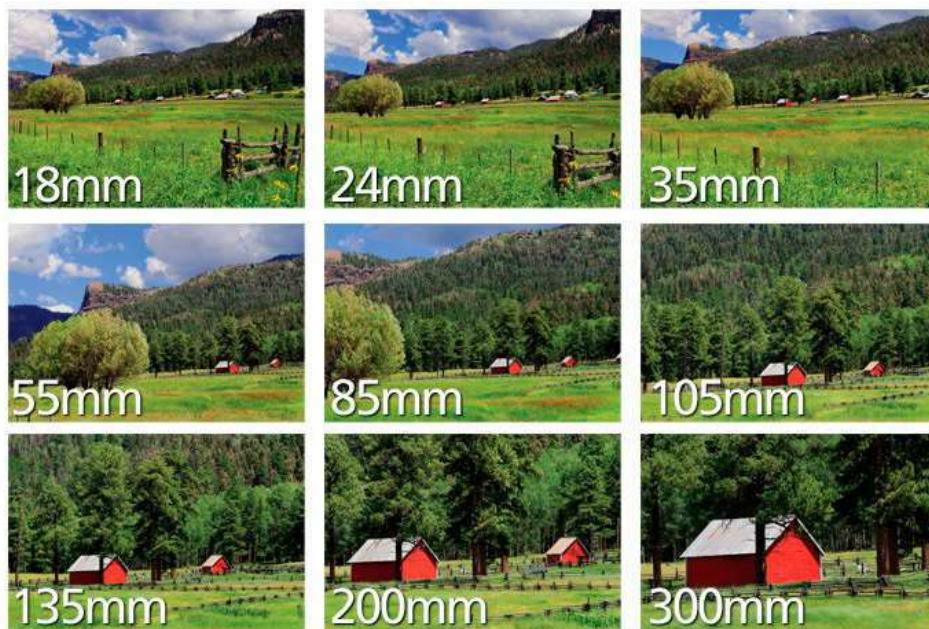


<https://www.dzoom.org.es/>

Ojo de Pez	Angulares	Normal / Estándar	Teleobjetivos	Súper Teleobjetivos
Distancia Focal	8mm	8 a 25mm	25 a 65mm	65 a 160mm
Ángulo de Visión	180°	110 a 60°	60 a 25°	25 a 10°

□ Ángulo de visión – distancia focal

- Escena captada con distintas distancias focales (en mm).



<https://www.nikon.com.mx/learn-and-explore/a/tips-and-techniques/entendiendo-la-distancia-focal.html>

- La **exposición** es a la cantidad de luz que recibe el sensor de imagen (en fotografía digital) o el material fotosensible (en fotografía química) para que se forme una imagen.
- **Depende de varios factores que debemos controlar:**
 - El **nivel de iluminación** recibido por el sensor.
 - El **tiempo** que el sensor está expuesto a dicha iluminación.
 - El resultado final también depende de la **sensibilidad** del sensor, que determinará la cantidad de luz que es capaz de captar.
- Para asegurar que la imagen tiene la exposición que deseamos, todos estos parámetros se pueden definir controlando ciertos elementos de la cámara (de forma automática o manual) como veremos a continuación.

□ La sensibilidad (ISO) (I):

- Es la **capacidad que tiene el sensor de la cámara para captar luz**.
- Tiene su origen en la **fotografía analógica** donde la película (el carrete) que se utilizaba era diferente:
 - En condiciones de buena luz se utilizaba un carrete “lento” con cristales de haluro de plata pequeños que proporcionaban mucho detalle y resolución.
 - Cuando había poca luz, era necesario utilizar carretes “rápidos”, con granos más grandes, que captaban mejor la luz, pero la imagen quedaba más granulosa.



<https://www.ds-km.com/2021/03/01/elegir-iso-asa-velocidad-de-pelicula/>

- El **obturador** de nuestra cámara es **un dispositivo mecánico que controla el tiempo durante el cual el sensor recibe luz**.
- El **tiempo de exposición** es el tiempo durante el cual el obturador está abierto, dejando pasar la luz al sensor.
- Habitualmente el parámetro con el que trabajamos es la **velocidad de obturación (Shutter Speed)**



$$\text{tiempo de exposición} = \frac{1}{\text{velocidad de obturación}}$$

<https://www.fotografiaesencial.com/blog/wp-content/uploads/2013/09/shutter-copia.jpg>

- Velocidades rápidas (poco tiempo de exposición) → se reduce el movimiento.
- Velocidades lentas (mucho tiempo de exposición) → imágenes movidas.

[https://www.dzoom.org.es/
para-que-sirve-la-velocidad-
de-obturacion/](https://www.dzoom.org.es/para-que-sirve-la-velocidad-de-obturacion/)



- Ejemplos: Fotografía tomada con distinta velocidad de obturación (tiempo de exposición)

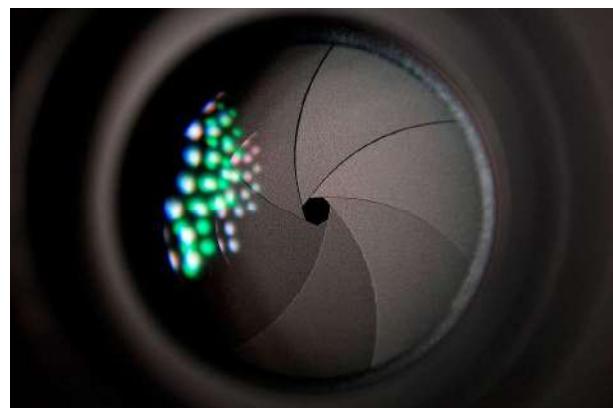


<http://fotopractica.com/2014/12/fotopractica-02-velocidad-de-obturacion/> [2021]

□ Diafragma

- Es un dispositivo que consta de unas palas que se abren y se cierran, dejando un orificio central que **permite pasar más o menos luz** hacia el sensor de nuestra cámara.
- Está en el **objetivo**, no en el cuerpo de la cámara.
- La **apertura del diafragma** normalmente aparece representada por el número **f**.

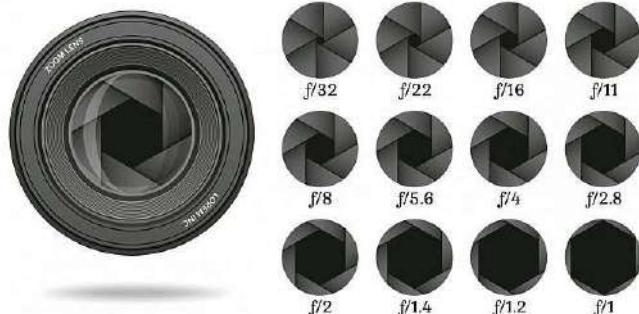
<https://www.dzoom.org.es/la-apertura-del-diafragma-en-fotografia-entiende-de-una-vez-por-todas-sus-implicaciones/>



□ Número f

- La **apertura de una lente** (del diafragma) se expresa con el **número f, N** en la fórmula, que es la **relación entre su longitud focal f y el diámetro D**
- **Notación:** Un nº f de N=16 suele indicarse como f/16 o F16.
- Efectos derivados de la abertura: cantidad de luz y nitidez
 - **Menor nº f: Gran abertura**
→ entra más luz en la cámara, pero menor profundidad de campo y nitidez.
 - **Mayor nº f: Pequeña abertura** → entra menos luz en la cámara, pero mayor profundidad de campo y nitidez.

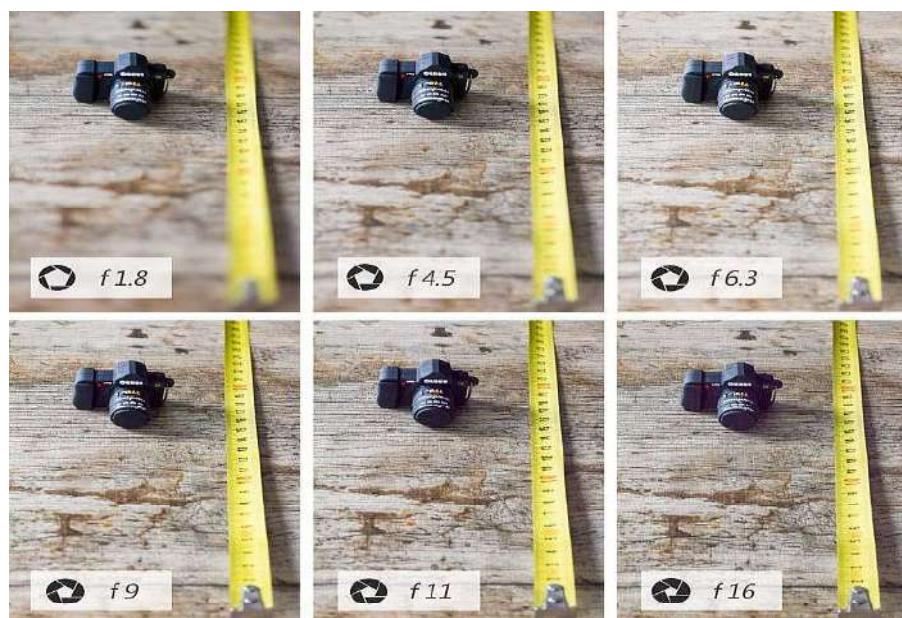
$$N = \frac{f}{D}$$



<https://www.dzoom.org.es/el-misterioso-significado-del-numero-f/>

Ejemplo profundidad de campo

- **Ejemplo:** fotografías tomadas con distinta apertura de diafragma (nº f) donde se aprecia su influencia en la profundidad de campo



<https://www.dzoom.org.es/profundidad-de-campo/>

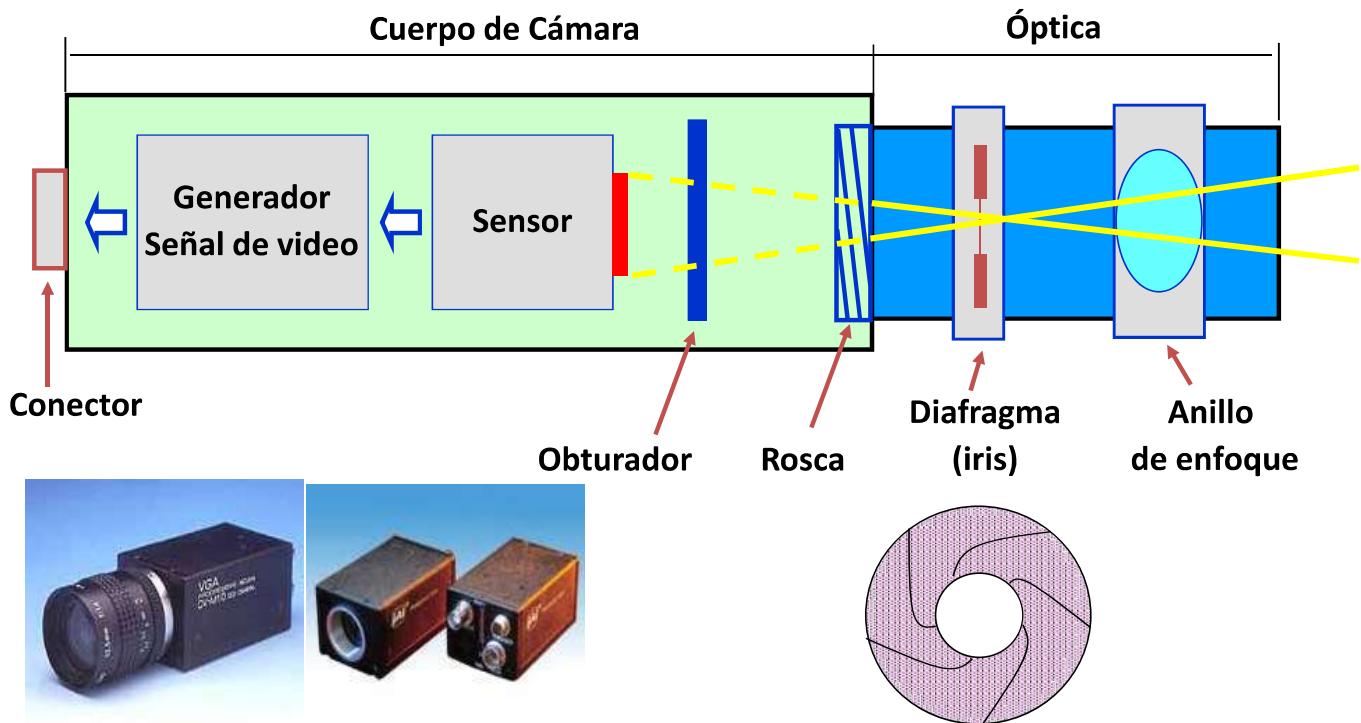
Triángulo de la Exposición



<https://www.dzoom.org.es/curso-de-fotografia-entendiendo-la-apertura/>



4. Cámaras



Cámaras - Sensores

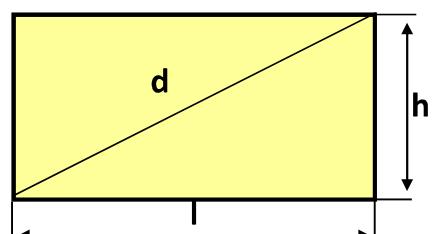


Canon CMOS
sensor,
EOS 400D

Sony CCD
sensor,
Nikon D2X

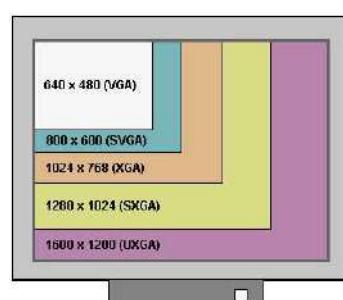
Sensor	1"	2/3"	1/2"
d (mm)	15.88	10.99	8.00
l (mm)	12.70	8.80	6.40
h (mm)	9.53	6.60	4.80

Tamaños típicos CCD



Clasificación

- Segundo la tecnología: CCD o CMOS.
- Segundo su formato: de Área, Lineales, TDI (Time Delay of Integration: tipo de CCD para captar imágenes de objetos móviles con poca luz).
- Segundo la Resolución: VGA, SVGA, XGA, SXGA...
- Segundo el Color:
 - Blanco y negro (un elemento CCD o CMOS por pixel)
 - Color (tres elementos CCD o CMOS por píxel (R,G,B))
- Segundo la Espectro: Visible, IR, Térmico, UV...
- Segundo la Salida: Analógica o digital.





5. Adquisición y geometría de formación de imágenes

Elementos que se incluyen en la adquisición y formación de imágenes

Escena  Iluminación  Óptica  Sensor

Señal  Digitalización  Representación digital

- Escena:** Realidad (espacio 3D).
- Iluminación:** Ilumina la escena (esta estrechamente relacionada con el tipo de sensor y los objetivos que se quieren alcanzar).
- Óptica:** Enfoca **luz** desde la escena sobre el sensor.
- Sensor:** Convierte **luz** en **energía eléctrica**.
- Señal:** Es una representación de la luz incidente como una energía eléctrica continua.
- Digitalizador:** Convierte señales continuas a señales discretas.
- Representación digital:** Representación final de la escena (realidad) en la memoria del ordenador/cámara digital.

- Una imagen puede ser representada por una función: $f(u,v)$: una matriz de números.
- (u,v) representan la localización de cada píxel en el plano imagen.
- El valor de $f(u,v)$ puede tener diferentes interpretaciones en diferentes tipos de imágenes. **Ejemplos:**

Imágenes de intensidad:

$f(u,v)$ = intensidad/nivel de gris.

Imágenes de distancia:

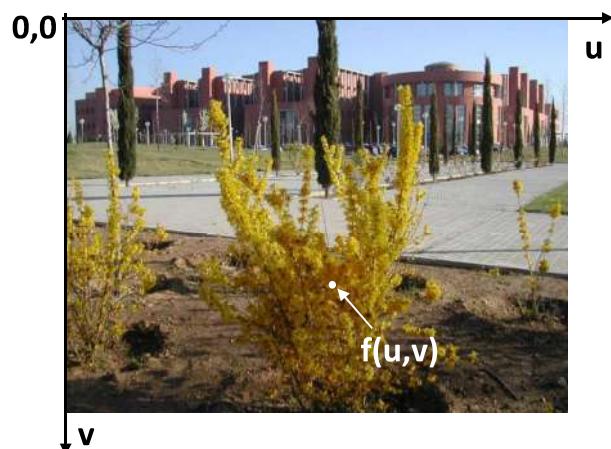
$f(u,v)$ = distancia desde la escena
al sistema de captación.

Imágenes en color:

$f(u,v) = \{f_r(u,v), f_g(u,v), f_b(u,v)\}$.

Vídeo:

$f(u,v,t)$ = secuencia temporal de imágenes.



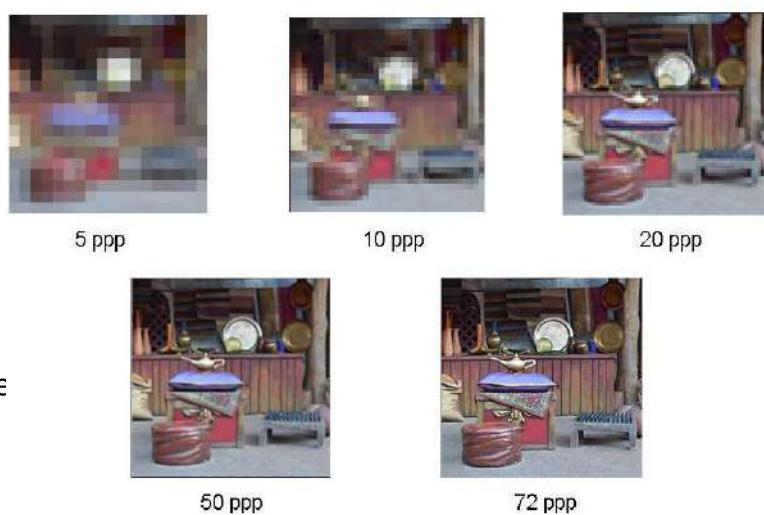
Factores que intervienen en la formación de imágenes

- **Digitalización:** en este caso es el paso en el que convertimos escenas con infinitos colores en matrices de números (representaciones digitales aproximadas).

Señal  Digitalización  Representación digital

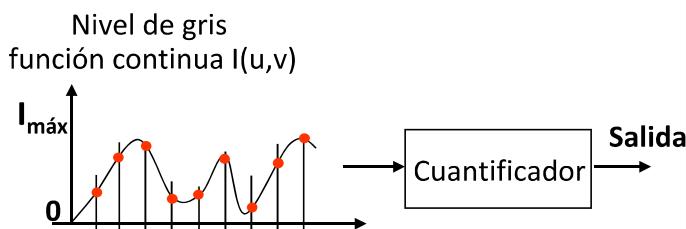
- 1^{er} paso en la digitalización, **muestreo:** tomar muestras/valores equiespaciadas.

- **Resolución espacial:** Depende del número de píxeles del dispositivo o, en caso de imágenes analógicas, del numero de muestras tomadas: valores típicos son MxN: 128x128, 256x256, 512x512, 1024x1024.
- Ejemplo de digitalización de una imagen con diferentes resoluciones espaciales suponiendo un patrón de muestreo cuadrado.



<http://musdatoslocos.blogspot.com/2016/02/diferencia-entre-resolucion-de-ppp.html>

- **Digitalización:** 2º paso, **cuantificación**, limitar el número de niveles de gris para poder representarlos numéricamente.
- Ejemplo en blanco y negro.



Nº bits para representac.	K: Num. valores distintos de salida
1	2
2	4
3	8
4	16
:	:
8	256

Imagen original:
Rampa lineal



Ejemplo de imagen con 8, 16, 32 y 64 niveles de gris de ECU dataset:



<https://link.springer.com/article/10.1007/s00521-017-3164-8/figures/4>

- **Digitalización:** ejemplo de cuantificación en color



K=2 (cada componente de color)



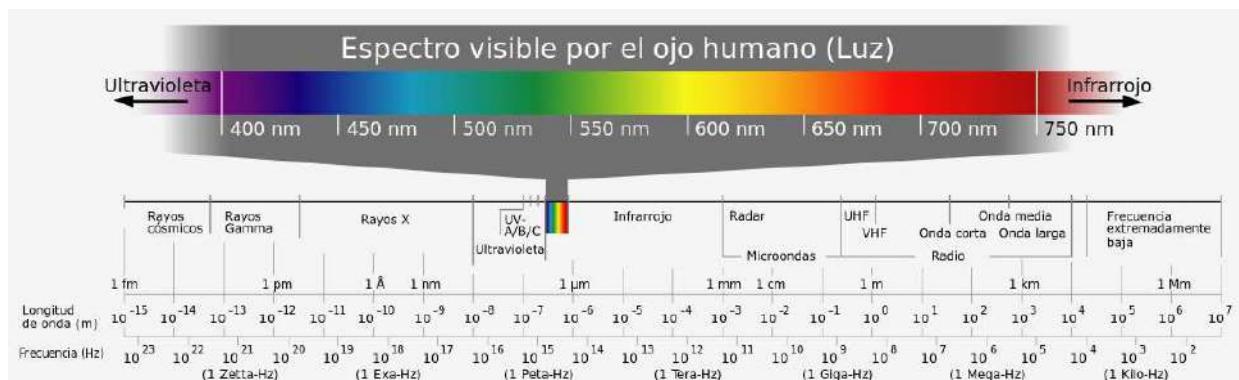
K=4 (cada componente de color)



10. Introducción al color

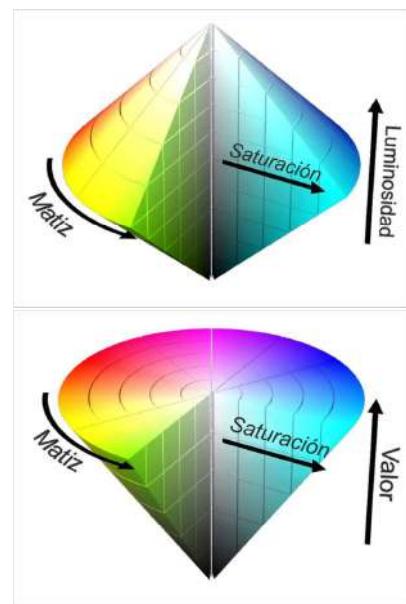
Color - ¿Cómo vemos?

- La **visión humana** está relacionada en especial con la **percepción del color, la forma, la distancia y las imágenes en tres dimensiones**.
- El **color** depende, en parte, de la longitud o longitudes de onda de las ondas luminosas **incidentes**, y en parte del estado del propio **ojo**, como ocurre en el **daltonismo**. Si la longitud de onda es superior o inferior a determinados límites no producen impresión visual.
- Las radiaciones electromagnéticas con longitud de onda entre 350 y 750 nanómetros (luz visible) estimulan los receptores humanos y producen la sensación de color.



Espectro electromagnético. Wikipedia.

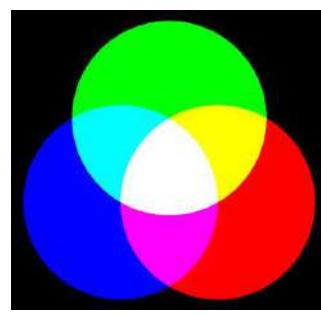
- Para **distinguir un color de otro** se pueden utilizar **diferentes características**. Las utilizadas con más frecuencia son **Brillo, Matiz (Tono) y Saturación**:
 - **Brillo**: incorpora la noción cromática de **luminosidad** (sensación que indica si un área está más o menos iluminada).
 - **Matiz**: Está asociado con la **longitud de onda dominante** en la mezcla de longitudes de onda de luz (sensación que indica si un área parece similar al rojo, amarillo, verde o azul o a una proporción de dos de ellos)
 - **Saturación**: Se refiere a la **pureza** depende de la distancia entre determinado color y la escala de grises; con la mayor saturación un color será más vivo, puro o colorido.
- El **matiz** y la **saturación** tomados conjuntamente se denominan **cromaticidad** (por tanto un color se puede definir por **brillo** y **cromaticidad**).



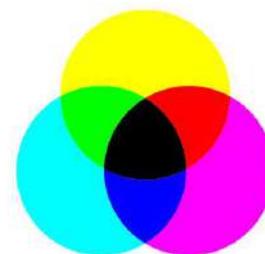
Doble cono de la coloración HSL y cono de la coloración HSV.
Wikipedia. Maulucioni, basado en un trabajo previo de SharkD.

Color - Espacios de color

- **Espacio (o modelo) de color**: es una especificación de un sistema de coordenadas 3D y un sub-espacio dentro de dicho sistema donde cada color se representa por un punto.
- Los espacios de color son siempre **tridimensionales**. Los espacios más frecuentes son:
 - **Red/Green/Blue (RGB)**: Síntesis aditiva (utilizado habitualmente en imágenes digitales: pantallas de ordenador y proyectores).
 - **Cyan/Magenta/Yellow (CMY)**: Síntesis sustractiva (utilizado en impresoras de color).
 - **Intensity/Chromaticity (YUV y YIQ)**: Se usan para transmisión de señales de televisión.
 - **Hue/Saturation/Value (HSV)**: Este espacio se usa típicamente en imágenes artísticas.
- En procesamiento de imágenes se utiliza RGB, YIQ, HSV (matiz, saturación, valor), HSI (matiz, saturación, intensidad). La I de YIQ y la I de HSI tienen significados distintos.

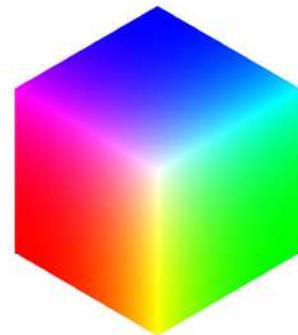


Síntesis aditiva. Wikipedia (Quark67, modificada por Monami)



Síntesis sustractiva. Wikipedia (Jorgelrm).

- El **espacio RGB** se basa en la combinación de las tres componentes espectrales primarias: R, G, B.
- Gráficamente se representa por un **cubo**.
- Cada componente toma valores entre 0 y 255, y un color concreto se define concatenando las tres componentes:



El blanco y el negro

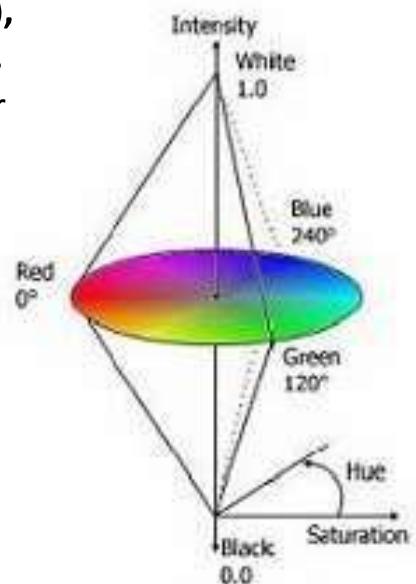
<u>Negro</u>	#000000	Los tres canales están al mínimo 00, 00 y 00
<u>Blanco</u>	#FFFFFF	Los tres canales están al máximo FF, FF y FF

Los tres colores básicos

<u>Rojo</u>	#ff0000	El canal de rojo está al máximo y los otros dos al mínimo
<u>Verde</u>	#00ff00	El canal del verde está al máximo y los otros dos al mínimo
<u>Azul</u>	#0000ff	El canal del azul está al máximo y los otros dos al mínimo

Las combinaciones básicas

<u>Amarillo</u>	#ffff00	Los canales rojo y verde están al máximo
<u>Cian</u>	#00ffff	Los canales verde y azul están al máximo
<u>Magenta</u>	#ff00ff	Los canales azul y rojo están al máximo
Gris claro	#D0D0D0	Los tres canales tienen la misma intensidad
Gris oscuro	#5e5e5e	Los tres canales tienen la misma intensidad



- Caracteriza el color en términos de **Matiz o tono (Hue)**, **Saturación o croma (Saturation)** y **brillo (Intensity)**. Es un modelo muy importante porque representa el color de forma similar a cómo lo sensa el ojo humano.
- Especial interés porque:
 - La componente de intensidad I, se puede separar de la información de color.
 - Las componentes tono H y saturación S, están estrechamente relacionadas con el modo en que los humanos perciben el color.
- **Hue** = Longitud de onda dominante o tono del color (ángulo entre 0º y 360º).
- **Saturation** = Saturación o pureza del color (entre 0 y 1).
- **Intensity** = Intensidad del color (entre 0 correspondiente al negro y 1 al blanco).

[http://iie.fing.edu.uy/
investigacion/grupos/
gti/timag/trabajos/
2010/maderas/](http://iie.fing.edu.uy/investigacion/grupos/gti/timag/trabajos/2010/maderas/)

- En muchas aplicaciones prácticas se plantea la necesidad de **realizar conversiones entre espacios de color** (en Rafael C. Gonzalez et al. 2004 se pueden encontrar varias).
- Ej: Las ecuaciones que permiten realizar la conversión entre RGB y HSI y viceversa son:

Conversión de RGB a HSI

$$H = \begin{cases} \gamma & \text{si } B \leq G \\ 360^\circ - \gamma & \text{si } B > G \end{cases} \quad \gamma = \cos^{-1} \left(\frac{R - \frac{1}{2}G - \frac{1}{2}B}{(R^2 + G^2 + B^2 - RG - GB - BR)^{1/2}} \right) \quad S = \left(1 - \frac{3\min(R, G, B)}{(R + G + B)} \right) \quad I = \frac{R + G + B}{3}$$

Conversión de HSI a RGB

$$\text{Sector RG } (0^\circ \leq H \leq 120^\circ): \begin{cases} B = I(1 - S) \\ R = I \left[1 + \frac{S \cos H}{\cos(60^\circ - H)} \right] \\ G = 3I - (R + B) \end{cases}$$

$$\text{Sector BR } (240^\circ \leq H \leq 360^\circ): H = H - 240^\circ \Rightarrow \begin{cases} B = I \left[1 + \frac{S \cos H}{\cos(60^\circ - H)} \right] \\ R = 3I - (G + B) \end{cases}$$

$$\text{Sector GB } (120^\circ \leq H \leq 240^\circ): H = H - 120^\circ \Rightarrow \begin{cases} R = I(1 - S) \\ G = I \left[1 + \frac{S \cos H}{\cos(60^\circ - H)} \right] \\ B = 3I - (R + G) \end{cases}$$

Sistemas de Visión Artificial

Grado en Ingeniería Computadores

*Tema 2: Técnicas básicas de procesamiento
de imágenes.*



*Autores: Sira Palazuelos, Luis M. Bergasa , Manuel Mazo,
M. Ángel García, Marisol Escudero, J. Manuel Miguel
Departamento de Electrónica. Universidad de Alcalá.*

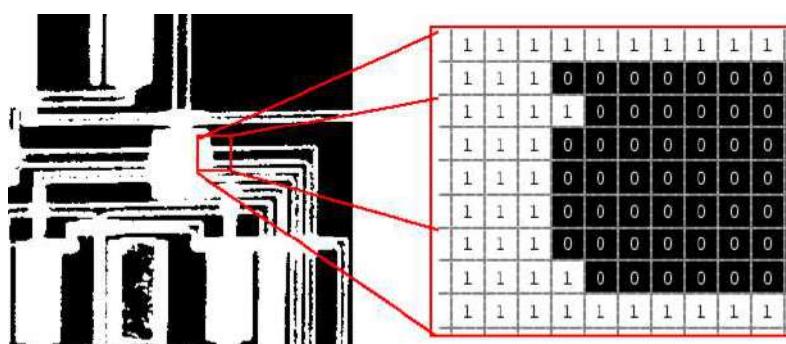
1

1. Representación de imágenes en Matlab
2. Mejora de imágenes
3. Transformaciones en el dominio del espacio
 - Transformaciones geométricas en imágenes
 - Histograma de una imagen y procesamiento basado en el histograma
 - Transformadas espaciales de la imagen
 - Ruido
4. Transformaciones en el dominio de la frecuencia

Representación de imágenes en Matlab

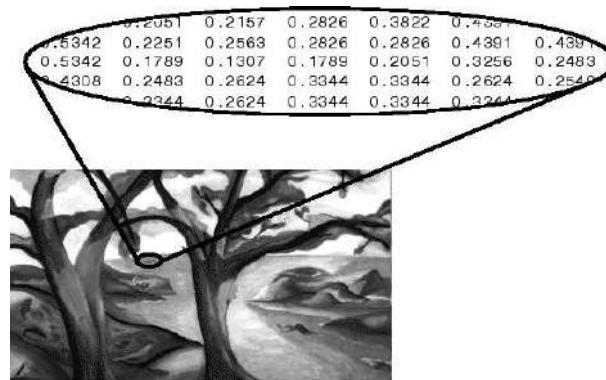
Representación de imágenes en Matlab

- **Formato de las imágenes.** Matlab puede trabajar con imágenes en los siguientes **formatos**: Graphics Interchange Format (.GIF), Tagged Image File Format (.TIFF), Bit Map Format (.BMP), ISO Standard Format (.JPEG , .JPG , .JPE), otros formatos (.PCX , .CGI , .PNG, etc.).
- Internamente Matlab puede **representar** las imágenes de diferentes maneras.
- Es imprescindible **conocer la representación** de la imagen antes de realizar el procesamiento: hay formas de representación que no indican directamente el color, algoritmos que no están disponibles para todas las representaciones, etc.
- **Representación de las imágenes (I)**
 - **Imágenes binarias:** La representación más sencilla: La imagen es una **matriz** $B = f(n,m)$ donde cada pixel puede valer 0 o 1: $f(i,j) = 0 \text{ o } 1$.



□ Representación de las imágenes (II)

- **Imágenes de intensidad (escala de grises):** cada imagen se representa con una matriz, $I = f(n,m)$ donde cada pixel es un **número**. En distintas imágenes estos números pueden ser de distinto tipo (reales o enteros) o tener distinto número de bits, dependiendo de la precisión con que estén codificados los colores. Ejemplos:
 - **Número entero (i)** con $0 \leq i \leq 255$ (0: negro; 255: blanco) o
 - **Número real (r)** con $0.0 \leq r \leq 1.0$ (0.0: negro; 1.0: blanco):



<https://es.mathworks.com/>

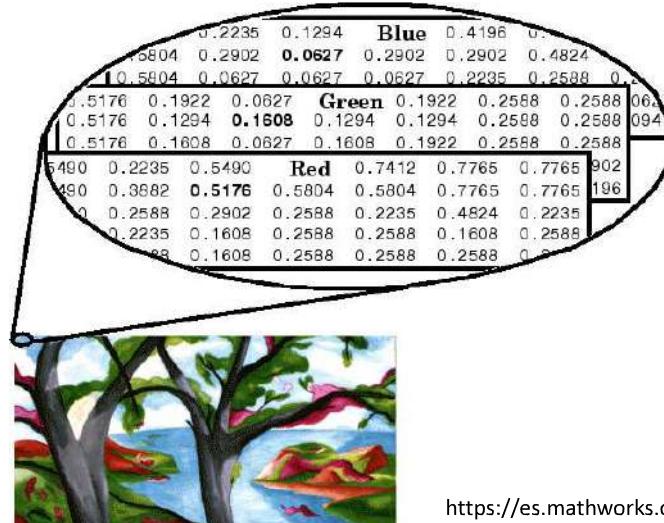
□ Representación de las imágenes (III)

- **Imágenes RGB:** Para representar una imagen de tamaño $N \times M$ utiliza **1 matriz con 3 componentes**: $R(n,m)$, $G(n,m)$, $B(n,m)$ → el tamaño total de la matriz es $N \times M \times 3$.

- En MATLAB, es posible extraer las 3 componentes:

- Si F es una imagen RGB: $FR = F(:, :, 1)$; $FG = F(:, :, 2)$; $FB = F(:, :, 3)$.

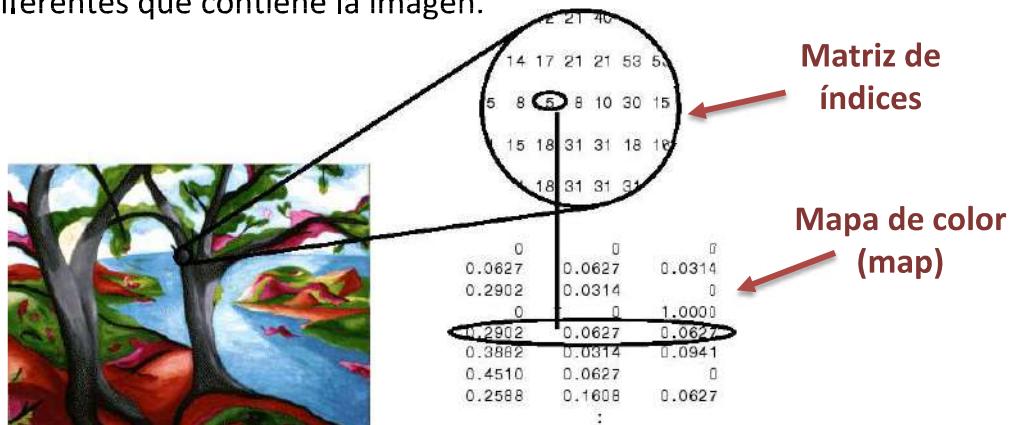
- **Problema de las imágenes RGB:** ocupan mucho espacio en memoria, ya que para cada pixel hacen falta 3 componentes. Si la imagen no tiene muchos colores diferentes, podemos optimizar la representación.



<https://es.mathworks.com/>

□ Representación de las imágenes (IV)

- Imágenes indexadas: se representan con una **matriz de índices (X)** y un **mapa de color (map)** que contiene todos los colores de la imagen.
- Los elementos de la imagen indexada NO son colores, son **índices** que apuntan al número de fila correspondiente del mapa de color.
- Cada fila del **mapa de color** contiene un número (en el caso de imágenes de intensidad) o un triplete (para imágenes RGB), que representan el color que debe tener ese pixel. El número total de filas del mapa de color, L, es el número de colores diferentes que contiene la imagen.

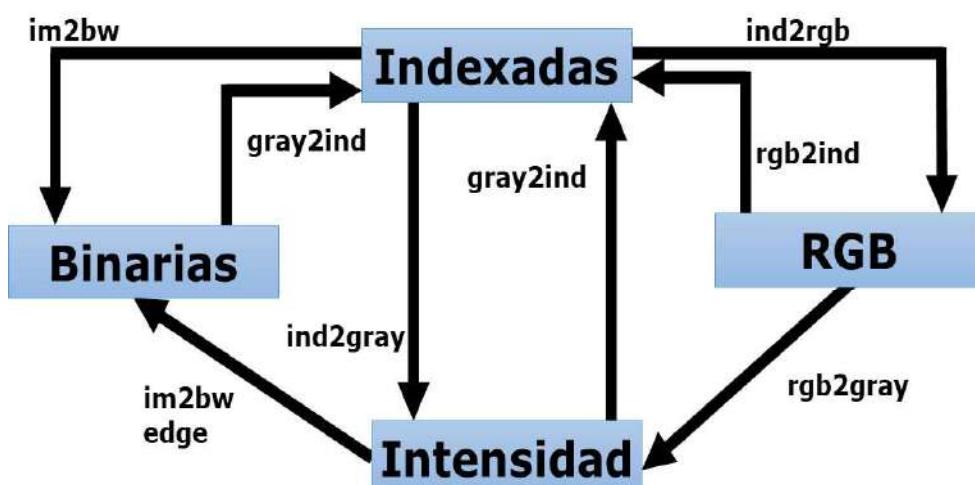


<https://es.mathworks.com/>

7

□ Representación de las imágenes (V)

- Si la imagen no está en el formato correcto para realizar el procesamiento, es necesario transformarla utilizando la **función de conversión** correspondiente:

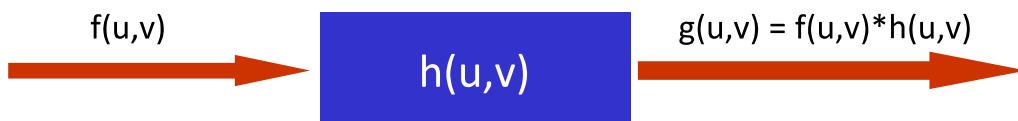


Mejora de imágenes

Mejora de imágenes

- Objetivo: cambiar las imágenes para **mejorar su “calidad”**.
 - Para la perspectiva humana.
 - Para posteriores operaciones de procesamiento.
- No existe una teoría general para definir la “calidad visual”.
 - Se asume: **si parece mejor, es mejor.**
- Dos alternativas:
 - Técnicas en el **dominio del espacio** → Operando directamente sobre los píxeles de la imagen: sobre el histograma, con máscaras...
 - Técnicas en el dominio **de la frecuencia** → Operando sobre la transformada de Fourier (por ejemplo) de la imagen.

□ Dominio del espacio



- u, v : coordenadas **espaciales** (plano imagen).
- $h(u,v)$: **respuesta al impulso** (respuesta a un punto de luz intenso caracterizado por la función delta de Dirac).
- $*$: operador **convolución**.
- Ejemplo de transformaciones en dominio del espacio:
 Imagen original Incremento de brillo Disminución de brillo



□ Dominio de la frecuencia (transformada de Fourier, del Coseno, Wavelet, etc.)



- p, q : **frecuencias espaciales** (frecuencia en la dirección de u y v respectivamente).
- $H(p,q)$: **función de transferencia** del filtro.
- Ejemplo de transformación en el dominio de la frecuencia:
 Imagen original Trasformada de la imagen Imagen de salida

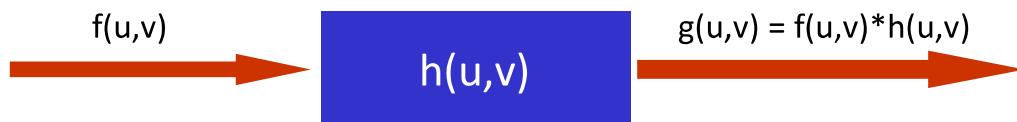


Mejora de imágenes

Mejora de imágenes

- Objetivo: cambiar las imágenes para **mejorar su “calidad”**.
 - Para la perspectiva humana.
 - Para posteriores operaciones de procesamiento.
- No existe una teoría general para definir la “calidad visual”.
 - Se asume: **si parece mejor, es mejor.**
- Dos alternativas:
 - Técnicas en el **dominio del espacio** → Operando directamente sobre los píxeles de la imagen: sobre el histograma, con máscaras...
 - Técnicas en el dominio **de la frecuencia** → Operando sobre la transformada de Fourier (por ejemplo) de la imagen.

□ Dominio del espacio



- u, v : coordenadas **espaciales** (plano imagen).
- $h(u,v)$: **respuesta al impulso** (respuesta a un punto de luz intenso caracterizado por la función delta de Dirac).
- $*$: operador **convolución**.
- Ejemplo de transformaciones en dominio del espacio:

Imagen original

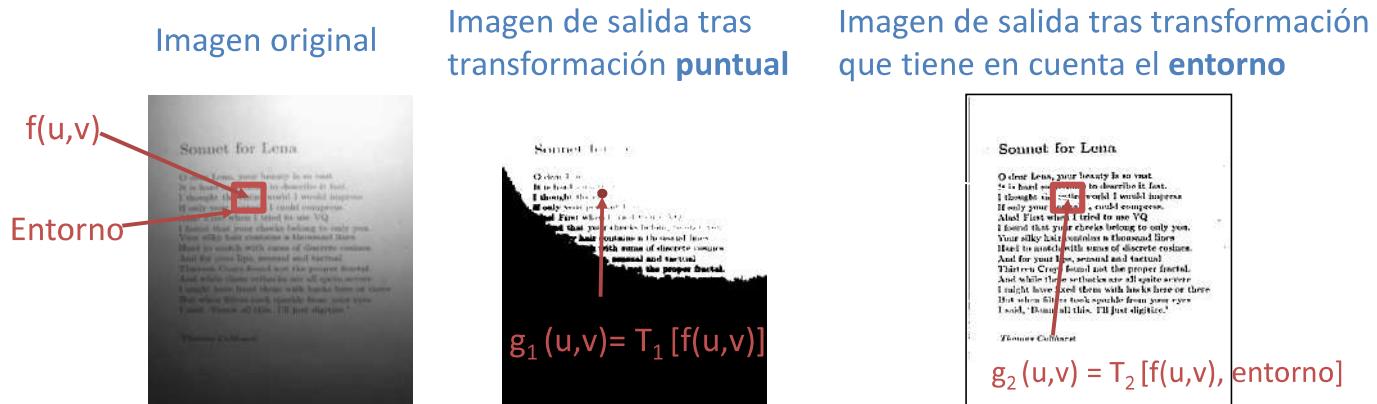
Incremento de brillo

Disminución de brillo



□ La transformación en el espacio T puede ser:

- **Puntual**: píxel a píxel: el valor final del pixel solo depende de su valor en la imagen original, no del entorno (p. ej.: aumento de brillo homogéneo a toda la imagen).
- **Área**: área local a píxel. Para calcular el valor final del pixel se tiene en cuenta el valor del entorno (utilizando máscaras).
- **Global**: imagen entera a píxel. Para calcular el valor final del pixel se tiene en cuenta toda la imagen (p.ej.: ecualización del histograma).

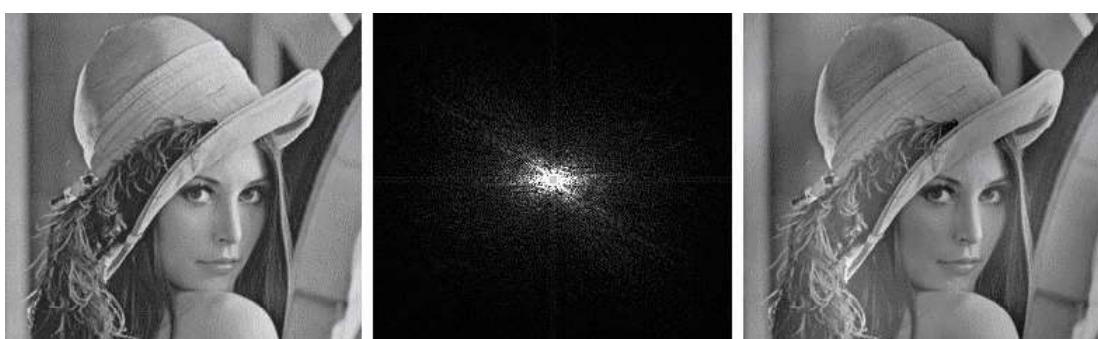


Dominio de la frecuencia (transformada de Fourier, del Coseno, Wavelet, etc.)



- **p, q: frecuencias espaciales** (frecuencia en la dirección de u y v respectivamente).
- **H(p,q): función de transferencia** del filtro.
- Ejemplo de transformación en el dominio de la frecuencia:

Imagen original Trasformada de la imagen Imagen de salida



Transformaciones geométricas en imágenes

- Las **transformaciones geométricas** se utilizan para resaltar/optimizar ciertas **zonas** o regiones dentro de una imagen (**regiones de interés / regions of interest, ROIs**).
- Para ello se realizan operaciones que modifican las **coordenadas espaciales** de la imagen: operaciones geométricas.
- Algunas de estas transformaciones son: **traslaciones, rotaciones/giros y homotecias/zoom**.

Imagen 1. Original



Imagen 2.
Recorte



Imagen 3.
Giro



Imagen 4.
Recorte

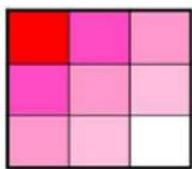


Imagen 5. Zoom: Final



<http://www2.diariomotor.com/imagenes/2012/10/posts/impuesto-de-matriculacion-01.jpg> [2021]

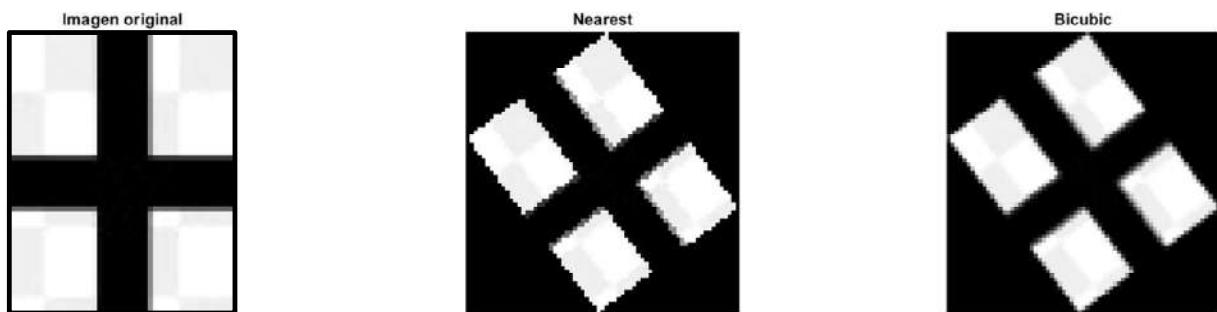
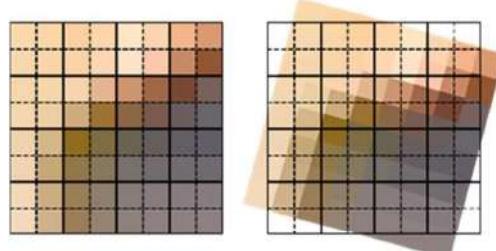
- El objetivo de una **operación geométrica** es **transformar** los valores de una imagen para que se vean como podrían observarse desde otro punto de vista.
- Las imágenes son **discretas** (entre dos píxeles no existen valores de intensidad) formando una rejilla, donde las coordenadas de cada celda son números enteros.
- Cuando se aplica la transformación (un desplazamiento, giro o zoom), los píxeles originales quedarían **desplazados** a una nueva posición, por ejemplo, para el caso de la traslación:



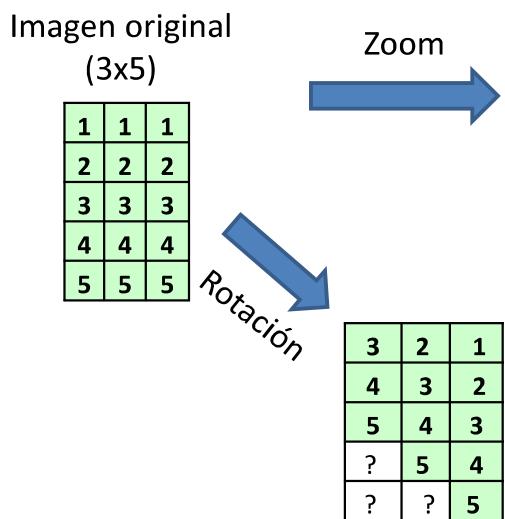
Ejemplo de:
[http://slideplayer.es/
slide/4333895/](http://slideplayer.es/slide/4333895/)

- En general, los píxeles de la imagen de origen **no** acaban en un único pixel concreto de la imagen de salida y en algunos casos pueden generarse **píxeles nuevos** (si se aumenta de tamaño una zona, por ejemplo).

Ejemplo de: http://www.kolor.com/wiki-en/Interpolation_and_benders [2021]



- En estos casos, es necesario utilizar un **algoritmo de interpolación** que determine el nivel de intensidad de los píxeles de la imagen final a partir de uno o varios que lo rodean. Se usa en giros, homotecias, etc...



?	?	?	?	?	?	?
?	1	?	1	?	1	?
?	?	?	?	?	?	?
?	2	?	2	?	2	?
?	?	?	?	?	?	?
?	3	?	3	?	3	?
?	?	?	?	?	?	?
?	4	?	4	?	4	?
?	?	?	?	?	?	?
?	5	?	5	?	5	?
?	?	?	?	?	?	?

Imagen final 2 (3x5)

¿Qué valor le damos a los píxeles nuevos?

- **Homotecia/zoom:** consiste en seleccionar una parte de la imagen (región de interés), separarla del resto de la imagen original y realizar un proceso de **expansión**.



- Una forma de expansión frecuente, por ejemplo para agrandar una imagen (zoom): **interpolación lineal** (valor medio entre **dos** píxeles).
- Permite pasar de imágenes de NxN a imágenes de $(2N-1) \times (2N-1)$ y se puede repetir las veces que se quiera.

5	3	5
2	12	4
6	8	10

Imagen original

5	4	3	4	5
2	7	12	8	4
6	7	8	9	10

Imagen con filas expandidas

5	4	3	4	5
3.5	5.5	7.5	6	4.5
2	7	12	8	4
4	7	10	8.5	7
6	7	8	9	10

Imagen con filas y columnas expandidas

- Otra forma de expansión: **promedio del entorno de vecindad.**

1^a. Expandir:

$$I = \begin{bmatrix} 2 & 6 & 4 \\ 8 & 10 & 2 \\ 4 & 6 & 8 \end{bmatrix} \rightarrow I_{\text{exp}} =$$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \mathbf{2} & 0 & \mathbf{6} & 0 & \mathbf{4} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \mathbf{8} & 0 & \mathbf{10} & 0 & \mathbf{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \mathbf{4} & 0 & \mathbf{6} & 0 & \mathbf{8} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

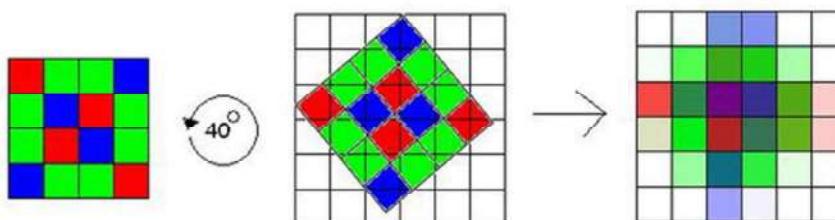
2^o. Promediar (convolución):

$$h = \begin{bmatrix} \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\ \frac{1}{2} & 1 & \frac{1}{2} \\ \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \end{bmatrix} \quad I_{\text{zoom}} = h * I_{\text{exp}}$$

$$\begin{bmatrix} 0.5 & 1 & 2 & 3 & 2.5 & 2 & 1 \\ 1 & 2 & 4 & 6 & 5 & 4 & 2 \\ - & - & - & - & - & - & - \\ - & - & - & - & - & - & - \\ - & - & - & - & - & - & - \\ - & - & - & - & - & - & - \\ - & - & - & - & - & - & - \end{bmatrix}$$

- Pero no todas las **transformaciones geométricas** son tan fáciles:

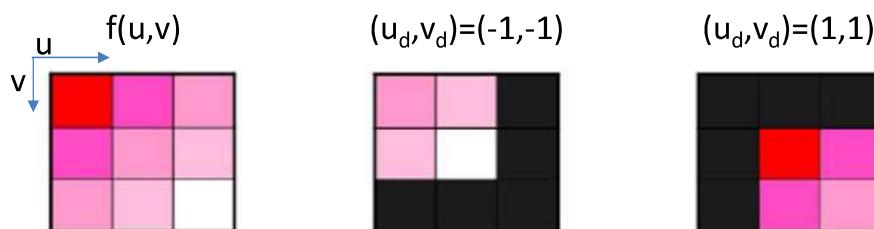
- En general, para llevar a cabo una transformación geométrica, es necesario **realizar 2 pasos**:
 1. Considerando la transformación, es necesario **calcular las coordenadas de la imagen de salida** en las que acabarán los píxeles de la imagen de entrada.
 2. A partir de dicha posición, se puede **calcular el valor de los píxeles** de la imagen de salida, utilizando algoritmos de interpolación si es necesario.



Ejemplo de: <http://www.codeproject.com/Articles/12230/Anti-Aliased-Image-Rotation>

1. Para calcular las coordenadas de la imagen de salida en las que acabarán los píxeles de la imagen de entrada, en caso de perspectivas, rotaciones y traslaciones, es muy conveniente poder representar las transformaciones y las coordenadas de los píxeles de forma **matricial**.
 - Concretamente, si los puntos se representan en **coordenadas homogéneas**, las transformaciones geométricas se podrán realizar como **producto de matrices**. De hecho, estas coordenadas permiten componer una sucesión de transformaciones en **una única matriz**.
 - Estas **coordenadas** agregan una componente a las coordenadas cartesianas:
- $(x, y) \rightarrow (x', y', W)$
- El punto representado en **coordenadas cartesianas** es el:
- $(x, y) = (x'/W, y'/W)$
- El valor de W es inicialmente es 1, y puede cambiar de valor al operar.
 - Ej: Dado el pixel en coordenadas cartesianas (2,3), se puede representar en coordenadas homogéneas como: (2, 3, 1), (4, 6, 2), (20, 30, 10)... todas ellas serán representaciones equivalentes del mismo punto.

- La **traslación**: dado un píxel $f(u,v)$, se desplaza (u_d, v_d) manteniéndose la intensidad $\rightarrow g(u_i+u_d, v_i+v_d) = f(u_i, v_i) \rightarrow g(u_f, v_f) = f(u_f-u_d, v_f-v_d)$.



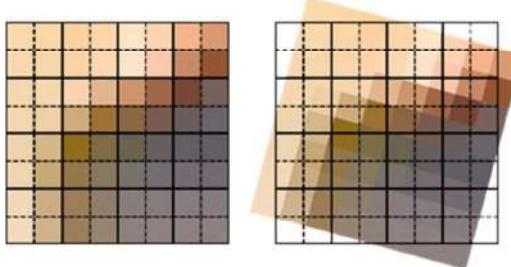
Ejemplo de:
[http://slideplayer.es/
 slide/4333895/](http://slideplayer.es/slide/4333895/)

- En coordenadas homogéneas:

$$\begin{bmatrix} u_f \\ v_f \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & u_d \\ 0 & 1 & v_d \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} \longrightarrow \begin{aligned} u_f &= u_i + u_d \\ v_f &= v_i + v_d \end{aligned}$$

- Donde u_f y v_f son las **coordenadas finales**, u_i y v_i las **iniciales**, y u_d y v_d el **vector desplazamiento**.

- Los **giros** se utilizan para producir efectos estéticos, y para simular la **rotación** de la cámara o del propio objeto.



Ejemplo de: http://www.kolor.com/wiki-en/Interpolation_and_blenders [2021]

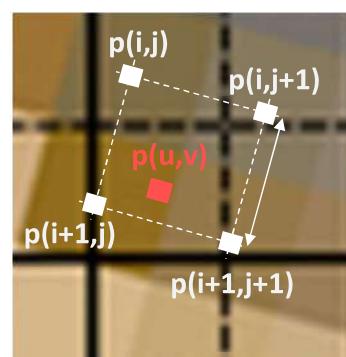
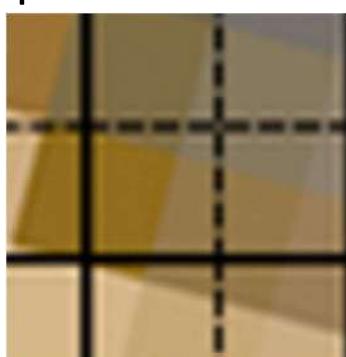
- Los parámetros necesarios para realizar la rotación son el **ángulo de giro** y las coordenadas del **centro de rotación**. La rotación cuando el centro está en el origen de coordenadas (0,0) viene dada por:

$$\begin{bmatrix} u_f \\ v_f \\ 1 \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix}$$

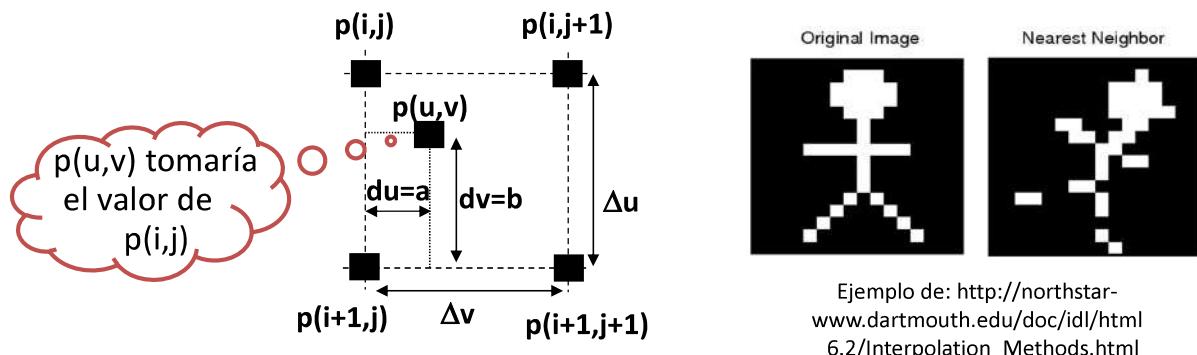
- La **rotación, respecto a un centro genérico** (u_d, v_d) viene dada por:

$$\begin{bmatrix} u_f \\ v_f \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & u_d \\ 0 & 1 & v_d \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -u_d \\ 0 & 1 & -v_d \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix}$$

- Una vez conocemos el destino de los píxeles, debemos calcular su valor definitivo en la imagen de salida: debemos calcular de qué color debe ser $p(u,v)$ en función de los colores de los píxeles cercanos la imagen original → esa operación se denomina **interpolación**.



- El tipo más fácil de **interpolación** es la **nearest**: el píxel $p(u,v)$ de la imagen final toma el **valor del píxel más cercano** (distancia euclídea) de los cuatro vecinos que le rodean.



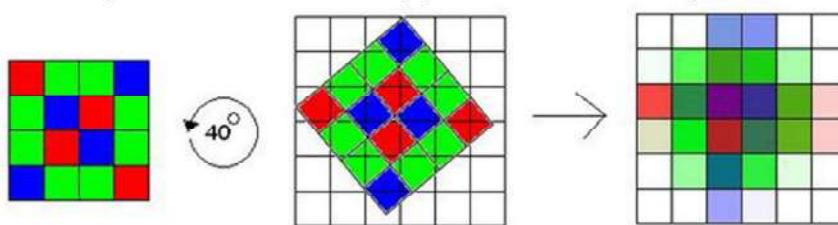
- La **interpolación bilineal** asigna a cada pixel el **valor medio ponderado** de las intensidades de los **cuatro** píxeles que le rodean.
- Los **factores de ponderación** vienen dados por la distancia entre el píxel y los del entorno:

$$p(u,v) = a_1 p(i, j) + a_2 p(i, j + 1) + a_3 p(i + 1, j) + a_4 p(i + 1, j + 1)$$

$$a_1 = (1 - du)(1 - dv); \quad a_2 = du(1 - dv)$$

$$a_3 = (1 - du)dv; \quad a_4 = du dv$$

- Da mejores resultados, pero tiene mayor coste computacional.



Ejemplo de: <http://www.codeproject.com/Articles/12230/Anti-Aliased-Image-Rotation>

- La **interpolación bicúbica**: El valor de un pixel interpolado es una **combinación** de los valores de los **16 píxeles más cercanos**.
- Este método produce una superficie mucho más **suave** que la interpolación bilineal. Para usar la interpolación bicúbica, la primera y segunda derivada de la superficie deben ser continuas.
- Es más **lenta** y ocupa **más memoria** que la interpolación bilineal.

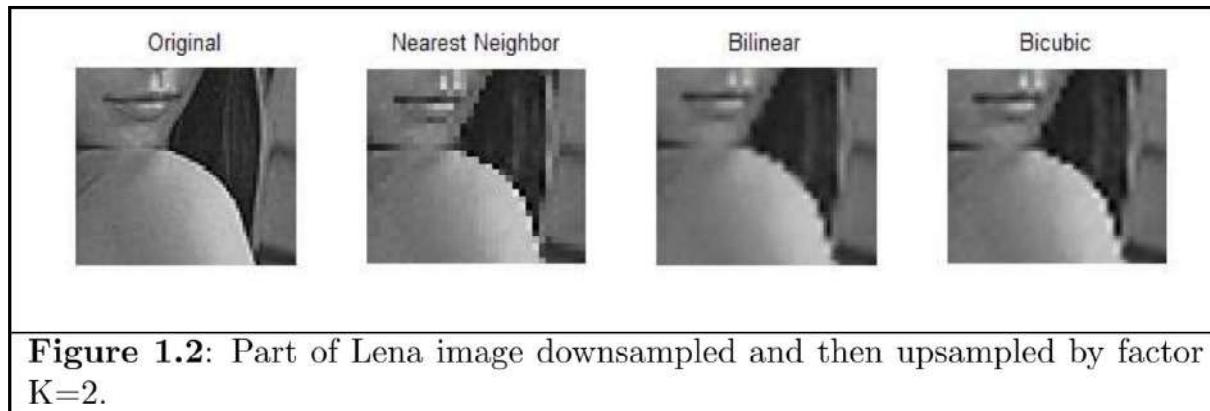


Figure 1.2: Part of Lena image downsampled and then upsampled by factor $K=2$.

<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.91.8726&rep=rep1&type=pdf>

Transformaciones geométricas en imágenes (continuación)

Transformaciones Geométricas

□ Jerarquía de transformaciones de coordenadas 2D

Transformation	Matrix	# DoF	Preserves	Icon
translation	$\begin{bmatrix} I & t \end{bmatrix}_{2 \times 3}$	2	orientation	
rigid (Euclidean)	$\begin{bmatrix} R & t \end{bmatrix}_{2 \times 3}$	3	lengths	
similarity	$\begin{bmatrix} sR & t \end{bmatrix}_{2 \times 3}$	4	angles	
affine	$\begin{bmatrix} A \end{bmatrix}_{2 \times 3}$	6	parallelism	
projective	$\begin{bmatrix} \tilde{H} \end{bmatrix}_{3 \times 3}$	8	straight lines	

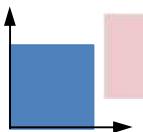
□ Transformación afín

- Es una transformación lineal donde M es una matriz 2×3 arbitraria.
- Al aplicarse, preserva el **parallelismo** de las rectas, pero **no los ángulos**.

$$P' = M_A * P$$

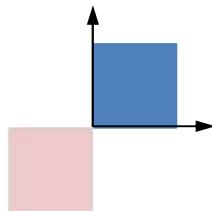
$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

□ Casos particulares:



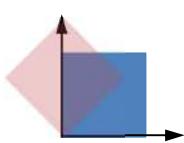
Translation

$$M_A = \begin{pmatrix} 1 & 0 & x_0 \\ 0 & 1 & y_0 \end{pmatrix}$$



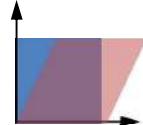
Mirror

$$M_A = \begin{pmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \end{pmatrix}$$



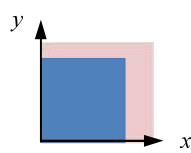
Rotation

$$M_A = \begin{pmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \end{pmatrix}$$



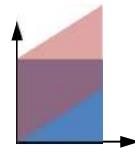
Shear x

$$M_A = \begin{pmatrix} 1 & A & 0 \\ 0 & 1 & 0 \end{pmatrix}$$



Scale

$$M_A = \begin{pmatrix} a & 0 & 0 \\ 0 & b & 0 \end{pmatrix}$$



Shear y

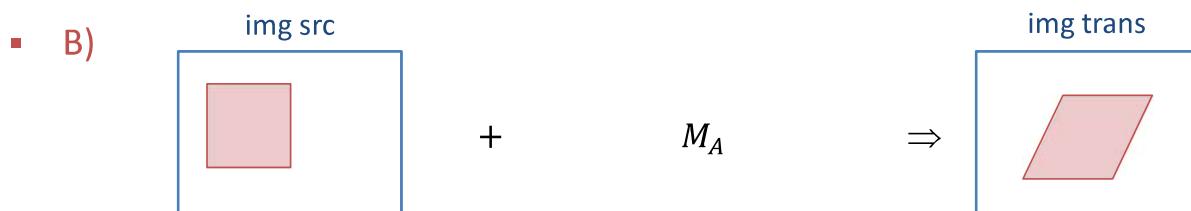
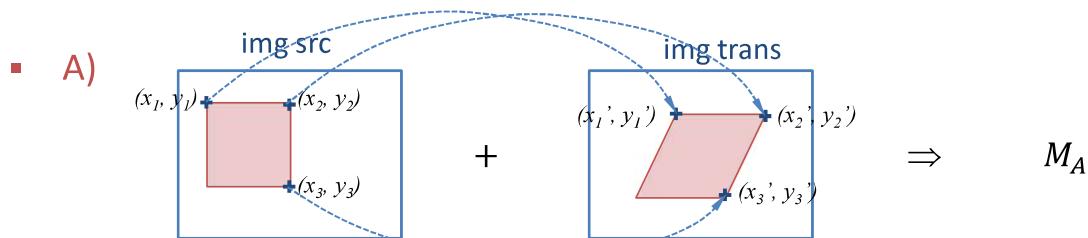
$$M_A = \begin{pmatrix} 1 & 0 & 0 \\ B & 1 & 0 \end{pmatrix}$$

□ Transformación afín

- Si tenemos 2 imágenes y de una puede obtenerse la otra aplicando una transformación afín, podemos obtener la matriz M_A de transformación conociendo la posición de **tres pares de puntos** en ambas imágenes.

$$P' = M_A * P$$

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

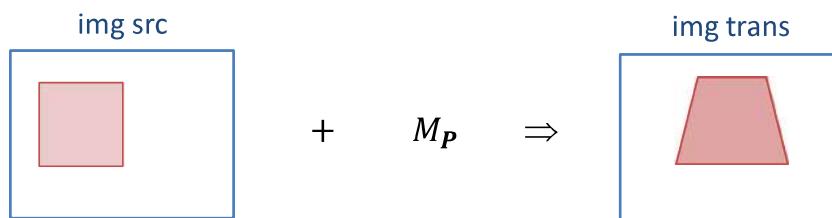


□ Transformación de Perspectiva (transformación proyectiva 3D a 2D):

- Es más general que la transformación afín.
- La proyección de perspectiva **preserva la linealidad**. La proyección de una línea 3D es una línea en 2D.
- **No necesariamente preserva el paralelismo** de las rectas.
- Generalmente $M_{33}=1$, por lo que se puede obtener la matriz de perspectiva a partir de **cuatro pares de puntos**.

$$P' = M_p * P$$

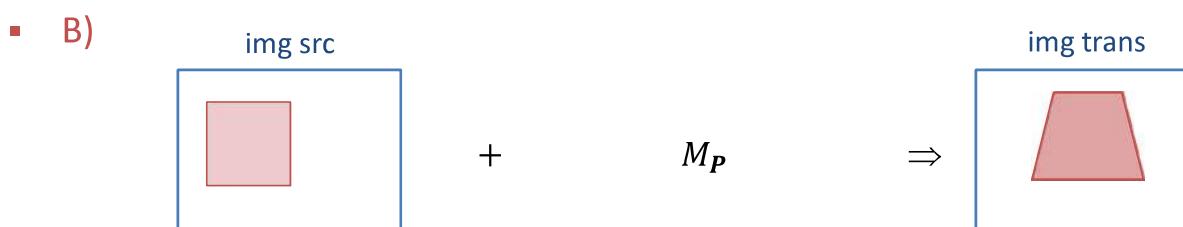
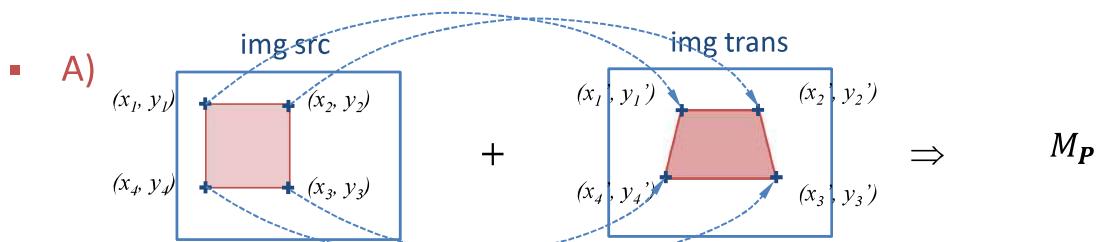
$$\begin{pmatrix} x' \\ y' \\ w \end{pmatrix} = \begin{pmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ M_{31} & M_{32} & M_{33} \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad \begin{cases} x' = \frac{M_{11}x + M_{12}y + M_{13}}{M_{31}x + M_{32}y + M_{33}} \\ y' = \frac{M_{21}x + M_{22}y + M_{23}}{M_{31}x + M_{32}y + M_{33}} \end{cases}$$



□ Transformación de Perspectiva

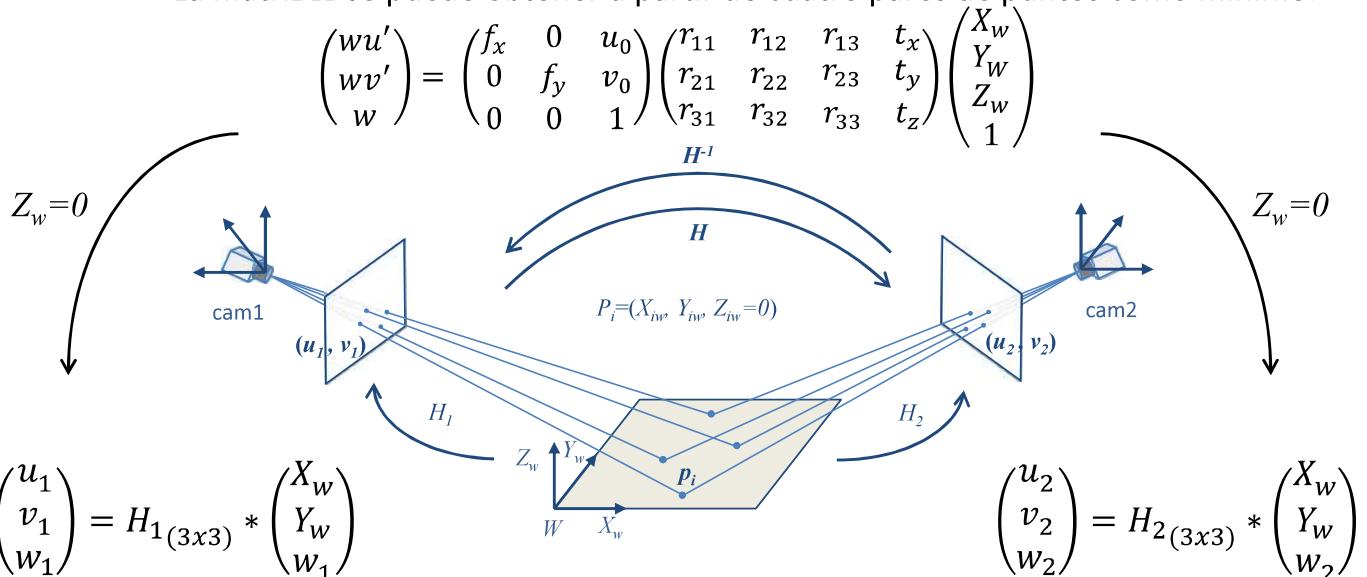
$$P' = M_p * P$$

$$\begin{pmatrix} x' \\ y' \\ w \end{pmatrix} = \begin{pmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ M_{31} & M_{32} & M_{33} \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad \begin{cases} x' = \frac{M_{11}x + M_{12}y + M_{13}}{M_{31}x + M_{32}y + M_{33}} \\ y' = \frac{M_{21}x + M_{22}y + M_{23}}{M_{31}x + M_{32}y + M_{33}} \end{cases}$$



□ Homografía

- Transformación proyectiva que determina una correspondencia entre **dos figuras geométricas planas**.
- Se aplica cuando hay una rotación pura ($t_x=t_y=t_z=0$) y figuras planas ($Z_w=0$).
- La matriz H se puede obtener a partir de cuatro pares de puntos como mínimo.



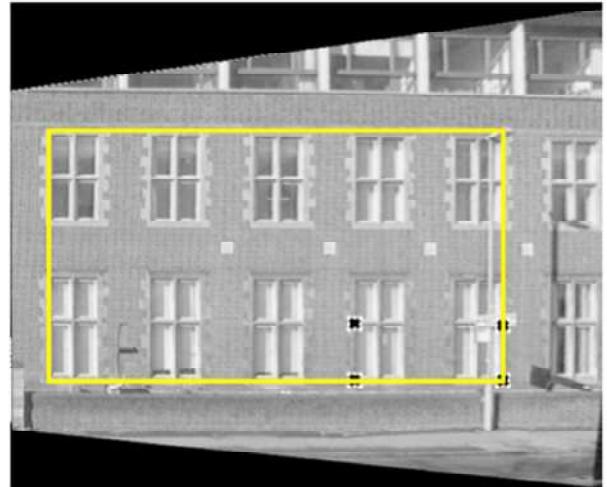
$$\left. \begin{array}{l} \begin{pmatrix} u_1 \\ v_1 \\ w_1 \end{pmatrix} = H_1 \cdot \begin{pmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{pmatrix} \quad \begin{pmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{pmatrix} = H_1^{-1} \cdot \begin{pmatrix} u_1 \\ v_1 \\ w_1 \end{pmatrix} \\ \begin{pmatrix} u_2 \\ v_2 \\ w_2 \end{pmatrix} = H_2 \cdot \begin{pmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{pmatrix} \end{array} \right\} \begin{pmatrix} u_2 \\ v_2 \\ w_2 \end{pmatrix} = H_2 \cdot H_1^{-1} \cdot \begin{pmatrix} u_1 \\ v_1 \\ w_1 \end{pmatrix}$$

$$\boxed{\begin{pmatrix} u_2 \\ v_2 \\ w_2 \end{pmatrix} = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix} \cdot \begin{pmatrix} u_1 \\ v_1 \\ w_1 \end{pmatrix}}$$

$$\left\{ \begin{array}{l} u'_2 = \frac{u_2}{w_2} \\ v'_2 = \frac{v_2}{w_2} \end{array} \right. \left\{ \begin{array}{l} u'_2 = \frac{h_{11} \cdot u_1 + h_{12} \cdot v_1 + h_{13} \cdot w_1}{h_{31} \cdot u_1 + h_{32} \cdot v_1 + h_{33} \cdot w_1} \\ v'_2 = \frac{h_{21} \cdot u_1 + h_{22} \cdot v_1 + h_{23} \cdot w_1}{h_{31} \cdot u_1 + h_{32} \cdot v_1 + h_{33} \cdot w_1} \end{array} \right.$$

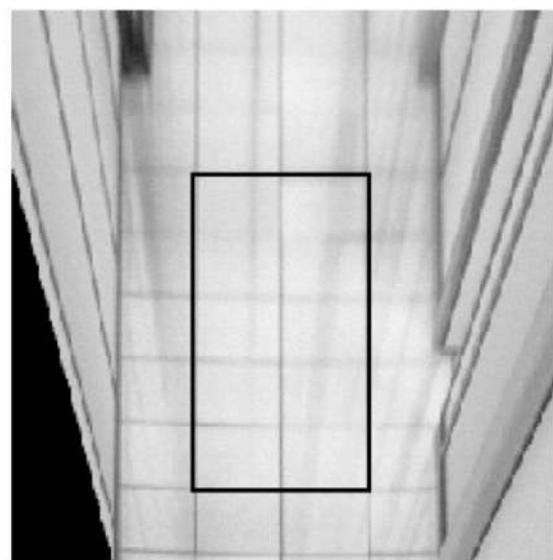
$$w_1 = 1 \quad \left\{ \begin{array}{l} -u'_2 \cdot u_1 \cdot h_{31} - u'_2 \cdot v_1 \cdot h_{32} - u'_2 \cdot h_{33} + h_{11} \cdot u_1 + h_{12} \cdot v_1 + h_{13} = 0 \\ -v'_2 \cdot u_1 \cdot h_{31} - v'_2 \cdot v_1 \cdot h_{32} - v'_2 \cdot h_{33} + h_{21} \cdot u_1 + h_{22} \cdot v_1 + h_{23} = 0 \end{array} \right.$$

- Aplicación de homografías para [eliminar](#) la distorsión de la perspectiva



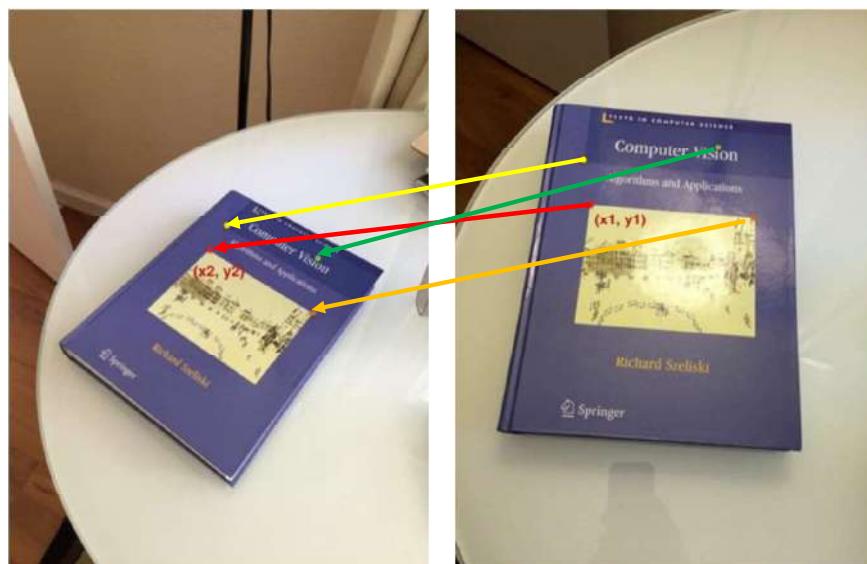
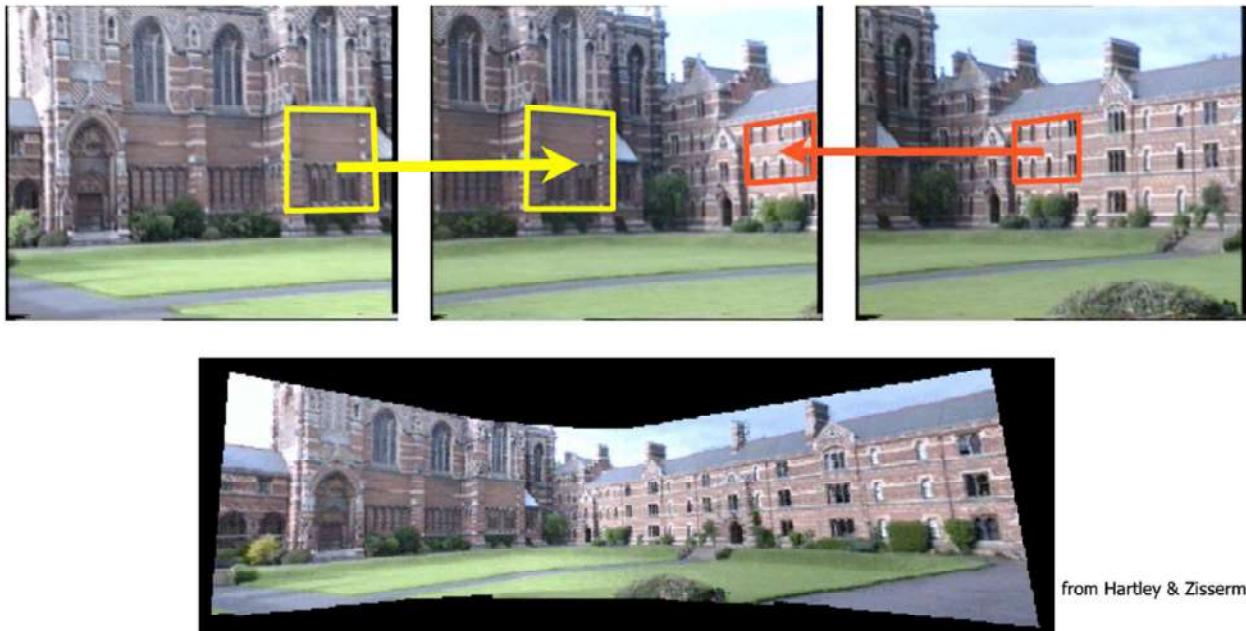
from Hartley & Zisserman

- Aplicación de homografías para crear una [vista de pájaro](#)



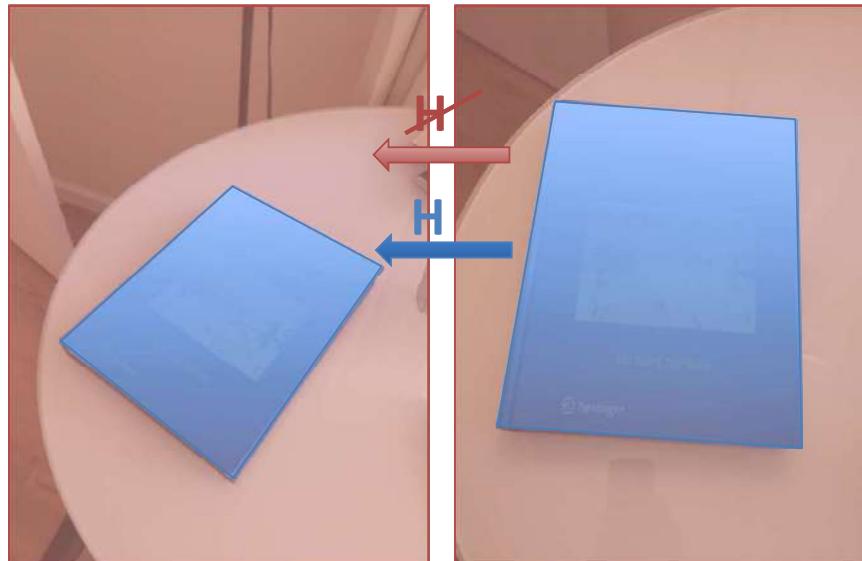
from Hartley & Zisserman

- Aplicación de homografías para crear imágenes panorámicas



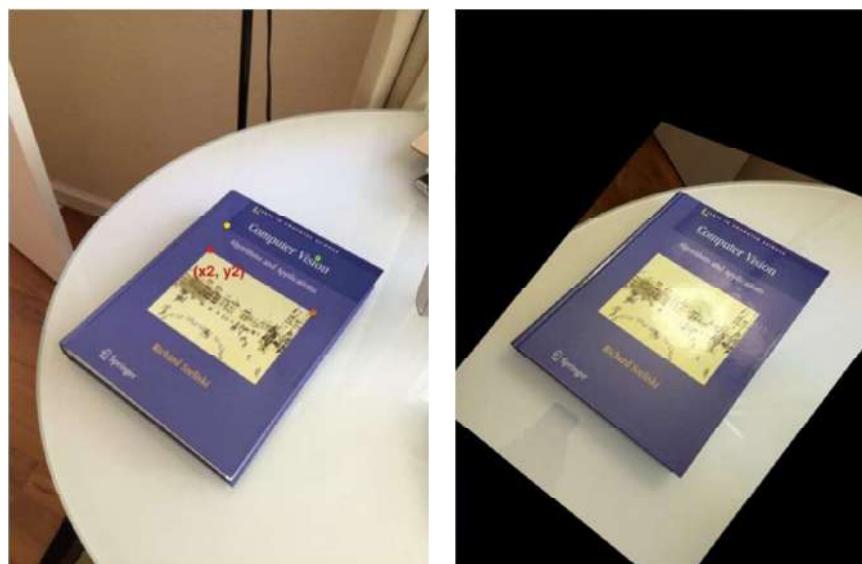
$$\begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} = H \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = \begin{bmatrix} h_{00} & h_{01} & h_{02} \\ h_{10} & h_{11} & h_{12} \\ h_{20} & h_{21} & h_{22} \end{bmatrix} \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix}$$

<https://www.learnopencv.com/homography-examples-using-opencv-python-c/>



$$\begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} = H \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = \begin{bmatrix} h_{00} & h_{01} & h_{02} \\ h_{10} & h_{11} & h_{12} \\ h_{20} & h_{21} & h_{22} \end{bmatrix} \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix}$$

<https://www.learnopencv.com/homography-examples-using-opencv-python-c/>



$$\begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} = H \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = \begin{bmatrix} h_{00} & h_{01} & h_{02} \\ h_{10} & h_{11} & h_{12} \\ h_{20} & h_{21} & h_{22} \end{bmatrix} \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix}$$

<https://www.learnopencv.com/homography-examples-using-opencv-python-c/>



$$\begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} = H \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = \begin{bmatrix} h_{00} & h_{01} & h_{02} \\ h_{10} & h_{11} & h_{12} \\ h_{20} & h_{21} & h_{22} \end{bmatrix} \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix}$$

<https://www.learnopencv.com/homography-examples-using-opencv-python-c/>



Visión Artificial & Sistemas de Visión Artificial

Tema 2. Geometría de la Cámara

Authors: Sira Palazuelos, Luis M. Bergasa , Manuel Mazo,
M. Ángel García, Marisol Escudero, J. Manuel Miguel
Department of Electronics. University of Alcalá.

Índice

1. Introducción
2. Modelo de la cámara
 - a. Modelo de las lentes delgadas
 - b. Modelo *pin-hole*
3. Geometría de la formación de la imagen
 - a. Sistemas de coordenadas
 - b. Coordenadas homogéneas
 - c. Cambios de referencia
4. Calibración de la Cámara



Introducción

Introducción

- La **imagen** captada por una cámara es una proyección del espacio tridimensional en un espacio bidimensional.
 - Se pierde información de la 3^a dimensión.
- A veces necesitamos **recuperar la tercera dimensión**, para poder situar los objetos en el espacio.
 - Posibles usos: robots para manipular objetos, interfaces de usuario, etc.
- Hay varias maneras de trabajar con 3D, con mayor o menor precisión:
 - Kinect.
 - Cámaras de tiempo de vuelo.
 - Recuperación de información 3D a partir de información 2D.



<http://www.ctim.es/site/>

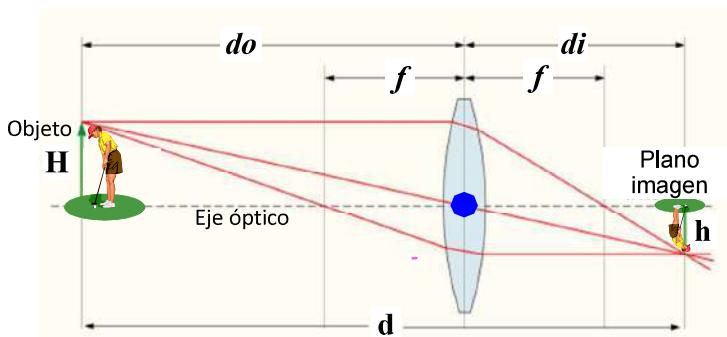
- Para poder reconstruir la información 3D a partir de la información de 2 imágenes debemos entender los pasos de la formación de las imágenes, para lo cual debemos :
 - Conocer las relaciones matemáticas que nos permiten establecer la relación entre las coordenadas físicas de un punto en el espacio tridimensional (X,Y,Z) y sus coordenadas en el plano imagen de la cámara (u,v) mediante:
 - **Modelo de la cámara**
 - **Geometría de formación de imágenes**
 - **Calibrar la cámara:** nos permite determinar los parámetros de la cámara a partir de medidas realizadas sobre puntos cuyas coordenadas espaciales son conocidas.



Modelo de la Cámara

- Determina la relación entre las coordenadas **3D** de los puntos de una escena y las coordenadas **2D** de sus proyecciones en la imagen.
- En el modelado intervienen parámetros:
 - **Ópticos**: tipo de lente, la distancia focal, distorsión, ángulo de apertura, etc.
 - **Fotométricos**: relacionados con la iluminación: intensidad, dirección de la iluminación, la reflectancia de los objetos.
 - **Geométricos**: tipo de proyección, posición y orientación de la cámara, perspectiva, etc.

- Ecuación de las **lentes delgadas**:

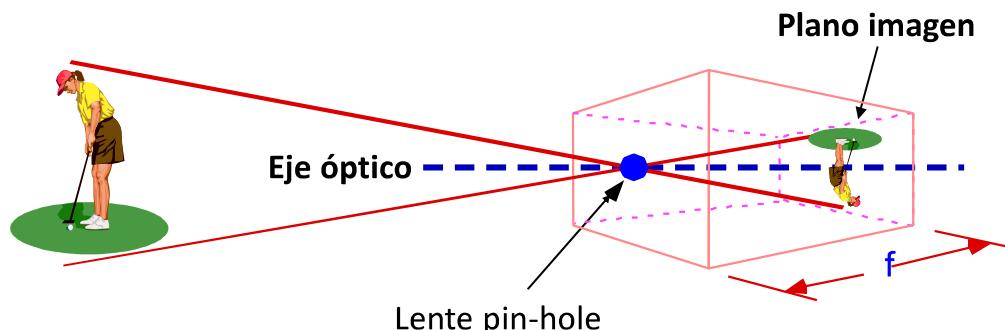


$$m = \frac{h}{H} = \frac{d_i}{d_o}$$

$$\frac{1}{d_o} + \frac{1}{d_i} = \frac{1}{f}$$

- f es la **distancia focal**: distancia entre el centro óptico de la lente y el foco o punto focal (lugar donde se enfocan los rayos que llegan a la lente paralelos al eje óptico).
- La escena se proyecta en una imagen 2D:
 - La imagen es invertida.
 - El tamaño se reduce.
 - Se pierde la información sobre la **profundidad** (distancia).

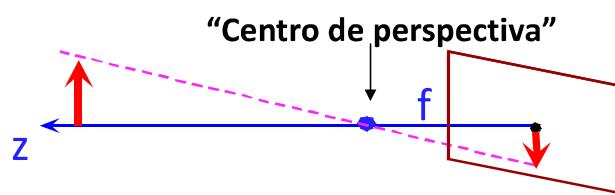
□ Modelo *pin-hole*



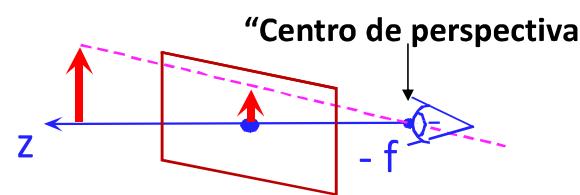
□ Simplificación con respecto a la ecuación de lentes delgadas:

- Se desprecia el efecto de la lente, considerando una cámara con una **apertura infinitesimal**.
- Asumir que el plano imagen está siempre a la **distancia focal f fija**.
- Se cumple cuando la distancia del objeto es mucho más grande de la distancia focal.

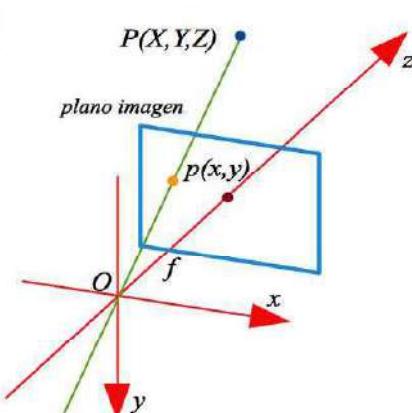
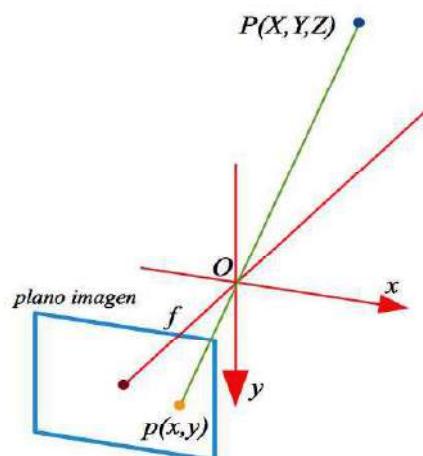
□ Se conoce como la “**proyección de perspectiva**”.



Centro de perspectiva $z = 0$



Plano imagen o de proyección $z = 0$



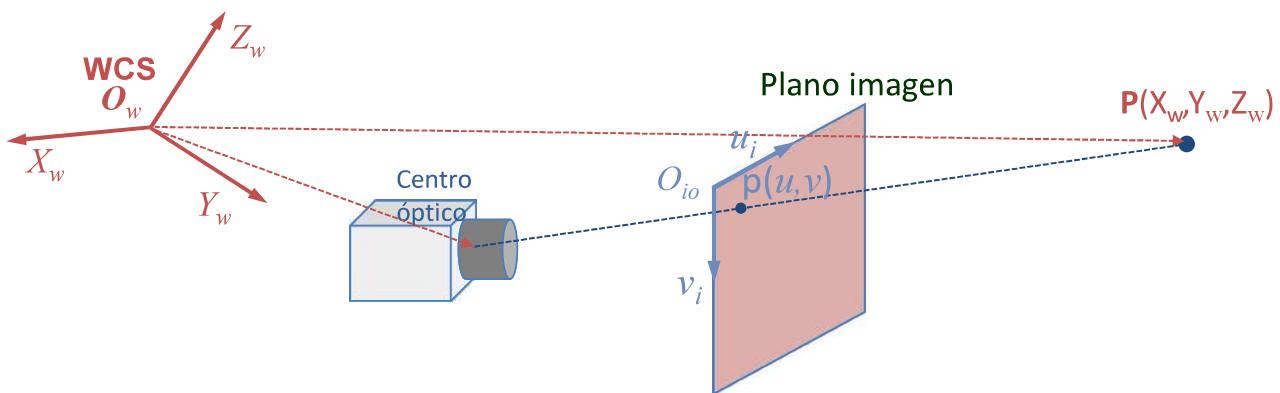


Geometría de la formación de la imagen

Geometría

□ Geometría de la formación de la imagen:

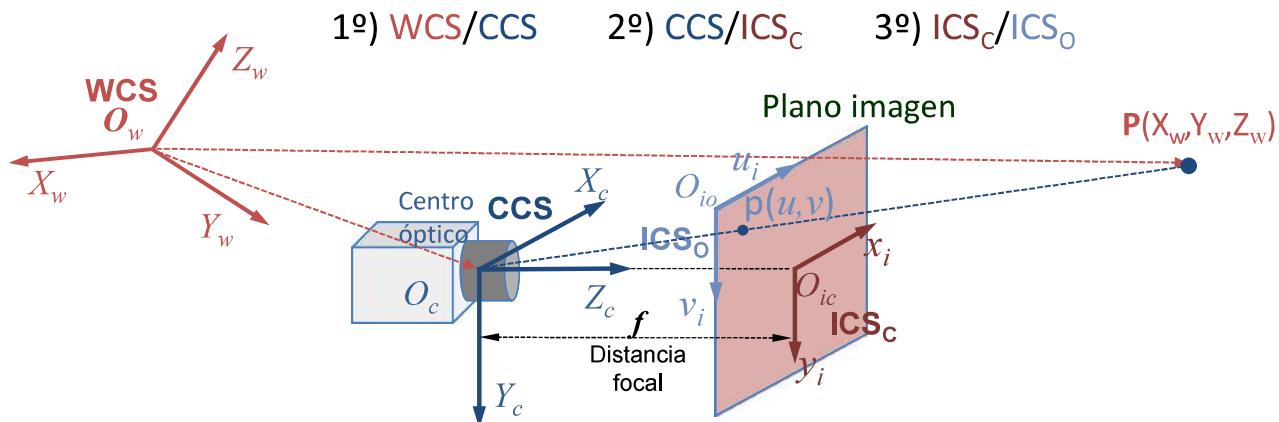
- Conocemos la posición de la cámara con respecto al sistema de coordenadas del mundo y la posición de un punto concreto en el mundo $P(X_w, Y_w, Z_w)$.
- Objetivo: **saber en qué píxel del plano imagen se va a proyectar.**



- Para conseguir el objetivo hay que realizar varios **cambios de coordenadas** (multiplicar por matrices sucesivas), que se pueden compactar en la **multiplicación por una única matriz**. Cuando conocemos esa matriz, podemos calcular en qué pixel se proyectará cualquier punto de la escena a partir de sus coordenadas en el mundo.

□ Sistema de coordenadas:

- La proyección del punto 3D sobre el plano imagen es la intersección entre la recta que une el **centro óptico (O_c)** con el **punto (P)** y el **plano imagen**.
- La proyección se obtiene tras realizar tres cambios de referencia:



WCS Sistema de coordenadas del mundo (O_w, X_w, Y_w, Z_w)

CCS Sistema de coordenadas de la cámara (O_c, X_c, Y_c, Z_c)

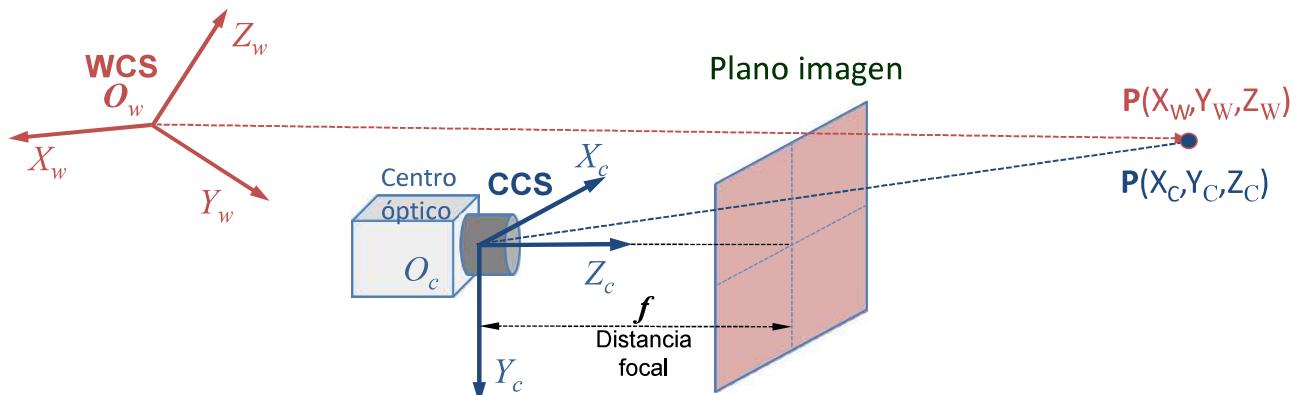
ICS_C Sistema de coordenadas de la imagen en el centro (O_{ic}, x_i, y_i)

ICS_O Sistema de coordenadas de la imagen en el origen (O_{io}, u_i, v_i)

Además, es necesario cambiar de unidades métricas a coordenadas pixélicas, para obtener el píxel concreto.

□ Cambio de referencia Mundo/Cámara (WCS/CCS)

- Se aplica una transformación lineal compuesta por una Translación $T_{3 \times 1}$ y una Rotación $R_{3 \times 3}$, multiplicando matrices. La multiplicación no es conmutativa (es distinto primero moverse y después girar que hacerlo al revés).

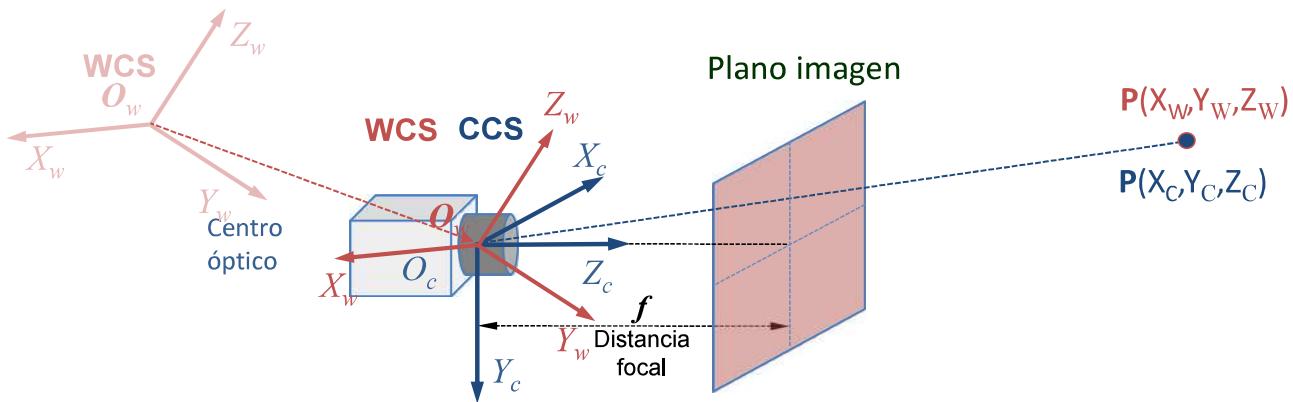


$$P_c = (M_1) \cdot P_w$$

$$P_c = \begin{bmatrix} R_{3 \times 3} & T_{3 \times 1} \\ 0 & 1 \end{bmatrix} \begin{pmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{pmatrix} = \begin{pmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{pmatrix}$$

□ Cambio de referencia Mundo/Cámara (WCS/CCS)

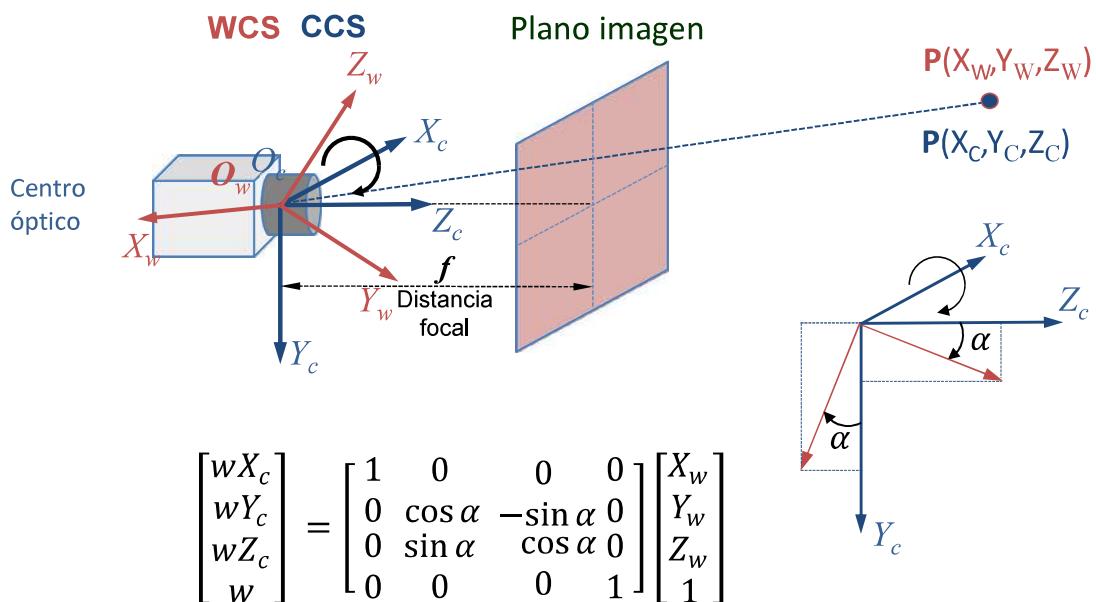
- Translación $T_{3 \times 1}$



$$\begin{bmatrix} wX_c \\ wY_c \\ wZ_c \\ w \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & T_x \\ 0 & 1 & 0 & T_y \\ 0 & 0 & 1 & T_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

□ Cambio de referencia Mundo/Cámara (WCS/CCS)

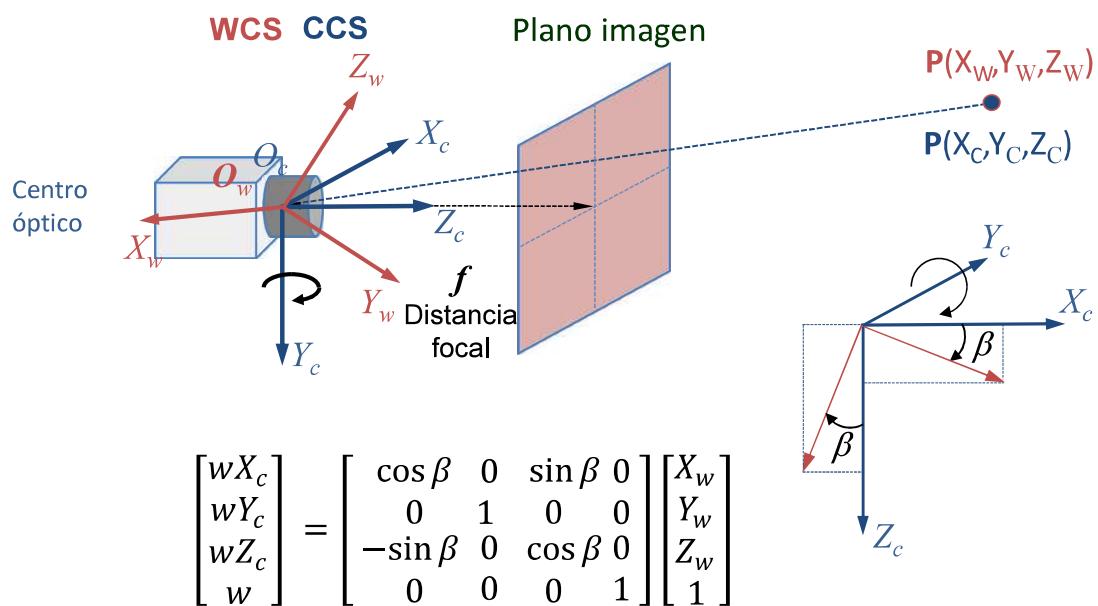
- X Rotación $R_X(\alpha)$



$$\begin{bmatrix} wX_c \\ wY_c \\ wZ_c \\ w \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha & 0 \\ 0 & \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

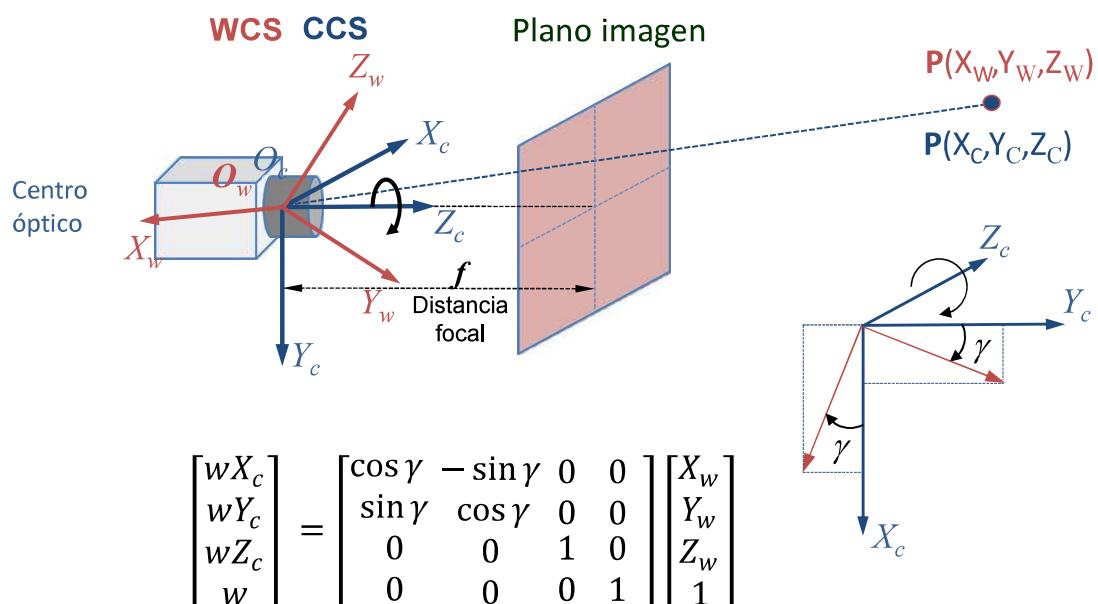
□ Cambio de referencia Mundo/Cámara (WCS/CCS)

- Y Rotación $R_Y(\beta)$



□ Cambio de referencia Mundo/Cámara (WCS/CCS)

- Z Rotación $R_Z(\gamma)$



□ Cambio de referencia Mundo/Cámara (WCS/CCS)

- La matriz completa, solo con la rotación:

$$R = R_X(\alpha) R_Y(\beta) R_Z(\gamma)$$

$$\begin{bmatrix} wX_c \\ wY_c \\ wZ_c \\ w \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha & 0 \\ 0 & \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \beta & 0 & \sin \beta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \beta & 0 & \cos \beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \gamma & -\sin \gamma & 0 & 0 \\ \sin \gamma & \cos \gamma & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

- La matriz completa realizando primero la traslación y después la rotación:

$$M = R_X(\alpha) R_Y(\beta) R_Z(\gamma) T$$

$$\begin{bmatrix} wX_c \\ wY_c \\ wZ_c \\ w \end{bmatrix} = \begin{bmatrix} \cos \gamma \cos \beta & -\sin \gamma \cos \beta & \sin \beta & T_x \\ \cos \gamma \sin \alpha \sin \beta + \sin \gamma \cos \alpha & \cos \gamma \cos \alpha - \sin \gamma \sin \alpha \sin \beta & -\sin \alpha \cos \beta & T_y \\ \sin \gamma \sin \alpha - \cos \gamma \cos \alpha \sin \beta & \sin \gamma \cos \alpha \sin \beta + \cos \gamma \sin \alpha & \cos \alpha \cos \beta & T_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

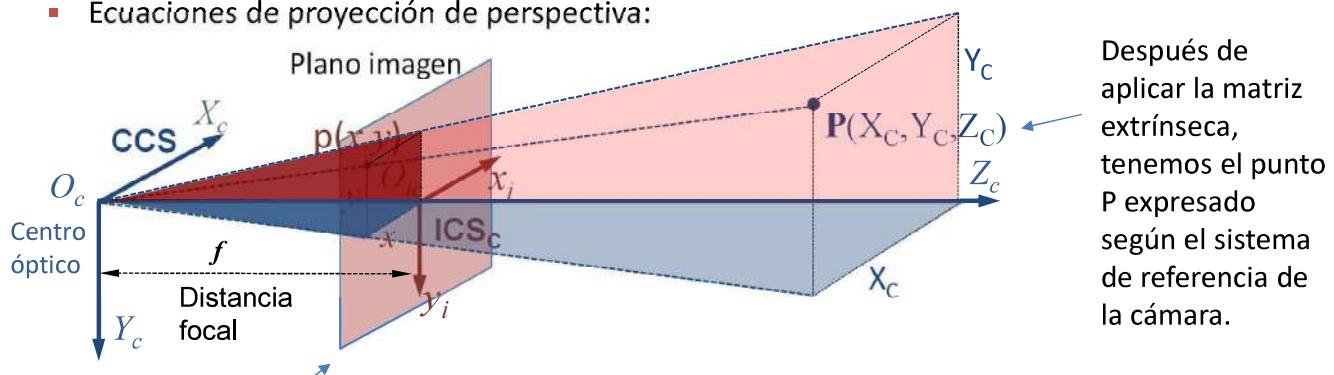
Matriz extrínseca: solo depende de la posición de la cámara con respecto al mundo (no de la cámara) → ¡Nos vale para cualquier cámara que esté en esa posición!

$$M_{ext} = \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{pmatrix}$$

12 incógnitas, aunque se puede simplificar a 6 porque la rotación sale de 3 ángulos → 3 incógnitas para rotación y 3 para traslación.

□ Cambio de referencia Cámara/Imagen (CCS/ICS_C)

- Proyectamos el punto en el sensor, en el plano imagen, perdiendo la tercera dimensión: el punto pasa de expresarse de 3 componentes en el mundo, a 2 componentes en el plano imagen.
- Ecuaciones de proyección de perspectiva:



En el plano imagen $z=f$
→ podemos calcular (x, y)
a partir de (X_c, Y_c, Z_c) por
semejanza de triángulos

$$\frac{y}{f} = \frac{Y_c}{Z_c}; y = f \frac{Y_c}{Z_c}$$

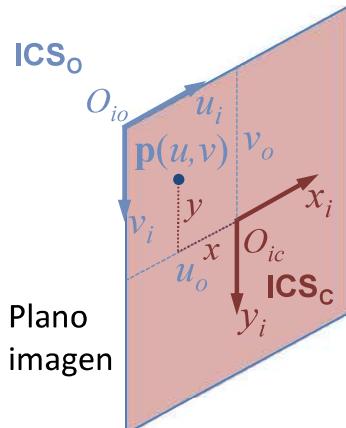
$$\frac{x}{f} = \frac{X_c}{Z_c}; x = f \frac{X_c}{Z_c}$$

$$\begin{bmatrix} wx \\ wy \\ w \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix}$$

Después de aplicar la matriz extrínseca, tenemos el punto P expresado según el sistema de referencia de la cámara.

□ Cambio de referencia (ICS_C/ICS_O) y de unidades (mm/píxel)

- La **digitalización** consiste en un cambio de **unidades** (mm/px) y un **offset** para trasladar el origen del sistema de coordenadas desde el centro del sensor al origen habitual de la imagen (esquina).

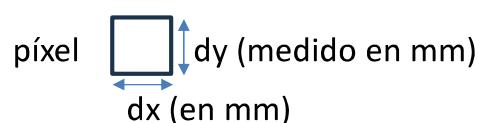


(u, v) coordenadas pixélicas: nuestro objetivo.

(x, y) coordenadas de p (proyección del punto P en el plano imagen) en mm referidas a O_{ic} .

(u_0, v_0) coordenadas pixélicas de O_{ic} (la intersección del centro óptico y el plano imagen) con respecto a O_{io} .

(dx, dy) dimensiones en mm de un píxel elemental de la imagen ($mm/píxel_x, mm/píxel_y$):



$$\begin{cases} u = \frac{x}{dx} + u_0 \\ v = \frac{y}{dy} + v_0 \end{cases}; \quad \begin{bmatrix} wu \\ wv \\ w \end{bmatrix} = \begin{bmatrix} 1/dx & 0 & u_0 \\ 0 & 1/dy & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} wx \\ wy \\ w \end{bmatrix}$$

- M_{int} : Matriz intrínseca**, calculada a partir de (u_0, v_0, f, dx, dy) , los llamados **parámetros intrínsecos**, que definen las características internas de la cámara → depende de cada cámara. Nos los puede dar el fabricante o los podemos medir.

$$M_{int} = \begin{pmatrix} 1/dx & 0 & u_0 \\ 0 & 1/dy & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} f/dx & 0 & u_0 \\ 0 & f/dy & v_0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{pmatrix}$$

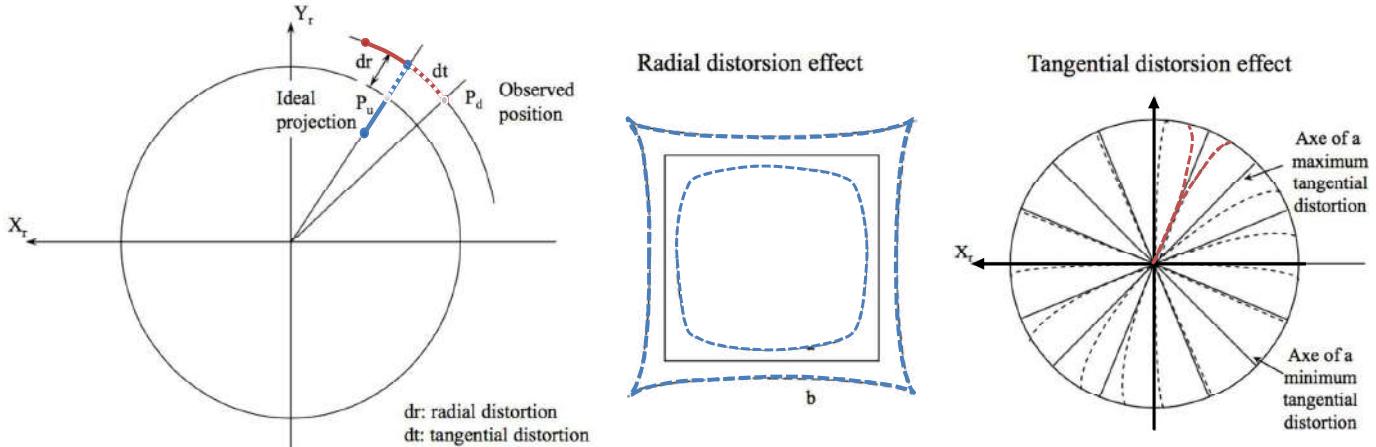
- M_{ext} : Matriz extrínseca**, compuesta por $(r_{(11, \dots, 33)}, t_{(x,y,z)})$ son los llamados **parámetros extrínsecos** que describen la transformación geométrica independientemente de la cámara: si colocamos otra cámara en la misma posición, tendrá la misma M_{ext} :

$$M_{ext} = \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{pmatrix}$$

- La expresión general del sistema completo es la **M (matriz de proyección)**, que nos permite calcular el píxel en el que se proyectará un punto cuando conocemos su posición en el sistema de coordenadas del mundo:

$$\begin{pmatrix} wu \\ wv \\ w \end{pmatrix} = M_{int} M_{ext} \begin{pmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{pmatrix} = \begin{pmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{pmatrix} \begin{pmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{pmatrix} = M_{(3x4)} \begin{pmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{pmatrix}$$

- Problema del modelo *pin-hole*: modela el caso ideal despreciando el efecto de la lente. Pero en realidad las lentes sí pueden influir en la formación de la imagen, por ejemplo, por las **distorsiones**. Puede haber varios tipos, por ejemplo:
 - **Distorsión radial**: cuando más lejos del centro de la lente estemos, más distorsión hay. Es muy común. Hay 2 tipos:
 - **Tipo barril**: cuanto más lejos del centro está el punto, más se acerca en la proyección.
 - **Tipo cojín**: cuanto más lejos está del centro, más se aleja.
 - **Distorsión tangencial**: los puntos no se proyectan en el mismo eje, sino se desplazan a la izquierda o a la derecha. Es menos común. Se produce cuando la lente no es paralela al sensor.



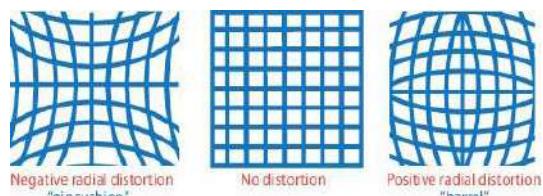
- ¿Cómo introducimos en el modelo el efecto de las distorsiones? Las corregimos antes de realizar el último cambio de referencia:

- P es un punto con coordenadas (X_c, Y_c, Z_c) referenciado a la cámara.
- x_n es la proyección normalizada en el plano imagen (modelo pinhole):

$$x_n = \begin{bmatrix} X_c/Z_c \\ Y_c/Z_c \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} \quad r = \sqrt{x^2 + y^2}$$

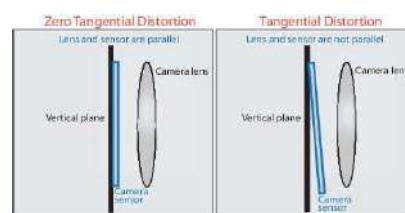
- La **distorsión radial** se modela multiplicando por un polinomio de coeficientes (k_1, k_2, k_3) , resultando (x_{rc}, y_{rc}) . (k_1, k_2, k_3) se estiman con una función que minimiza el error entre la posición que debería tener cada píxel y la que realmente tiene, utilizando gran cantidad de imágenes obtenidas con la misma lente (sistema sobredimensionado).

$$\left. \begin{aligned} x_{rc} &= x (1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \\ y_{rc} &= y (1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \end{aligned} \right\}$$



- La **distorsión tangencial** se modela sumando un polinomio de coeficientes (p_1, p_2) , resultando (x_{tc}, y_{tc}) . La expresión no solo depende de r , sino también de x e y .

$$\left. \begin{aligned} x_{tc} &= x + [2p_1 xy + p_2(r^2 + 2x^2)] \\ y_{tc} &= y + [p_1(r^2 + 2y^2) + 2p_2 xy] \end{aligned} \right\}$$



- Después de incluir la distorsión radial y tangencial las nuevas coordenadas normalizadas (x_d, y_d) se definen como:

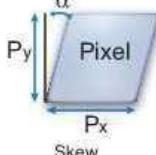
$$x_d = x (1 + k_1 r^2 + k_2 r^4 + k_3 r^6) + 2p_1 xy + p_2(r^2 + 2x^2)$$

$$y_d = y (1 + k_1 r^2 + k_2 r^4 + k_3 r^6) + p_1(r^2 + 2y^2) + 2p_2 xy$$

- Una vez aplicada la corrección de las distorsiones, las coordenadas (x_p, y_p) de la proyección del punto **P** en el plano imagen es:

$$\begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & sf_x & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_d \\ y_d \\ 1 \end{bmatrix}$$

- Donde s es **skew coefficient**, aplicado sólo si los ejes de la imagen no son perpendiculares (no suele ser necesario).



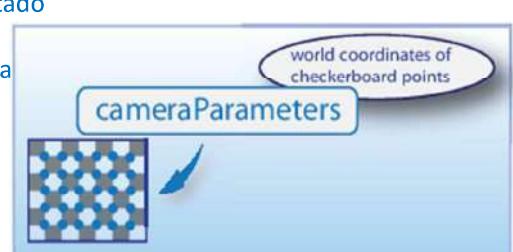
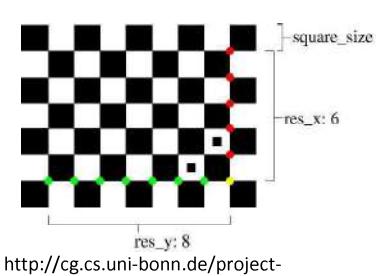
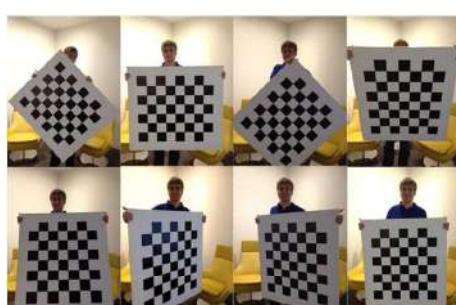
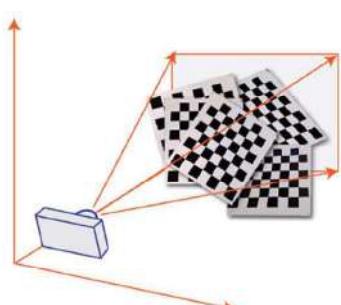
- Es decir, para poder calcular en qué pixel se va a proyectar un punto en el sistema de coordenadas del mundo se deben conocer **16 parámetros, 10 intrínsecos y 6 extrínsecos**.

$$\Phi = (u_0, v_0, f_x, f_y, k_1, k_2, k_3, p_1, p_2, s, \alpha, \beta, \gamma, t_x, t_y, t_z)$$

- El proceso por el cual se estiman los parámetros extrínsecos, intrínsecos y coeficientes de distorsión a partir de imágenes de un patrón de calibración se denomina **calibración de la cámara**.

Procedimiento:

- Captura imágenes de un **patrón de calibración conocido** (habitualmente un damero con dimensiones de los cuadrados conocidas) en diferentes orientaciones relativas a la cámara.
- Calibración:** calculamos las posiciones que debe tener cada punto en la imagen, comparamos con la posición en la que realmente se proyecta. Lo hacemos con gran cantidad de puntos y calculamos los parámetros para que el error sea mínimo.
- Evaluar los resultados de la calibración.



Histograma de una imagen y procesamiento basado en el histograma

Histograma de una imagen *Definición*

- Una **imagen** muestra la **distribución espacial** de los niveles de gris.
- El **histograma** de una imagen descarta la información espacial y muestra la frecuencia de ocurrencia de los valores de gris: **el número de veces que aparece en la imagen cada nivel de gris.**
- Una imagen tiene un solo histograma, pero un histograma se puede corresponder con infinitas imágenes.

Imagen: $M \times N = 6 \times 6$

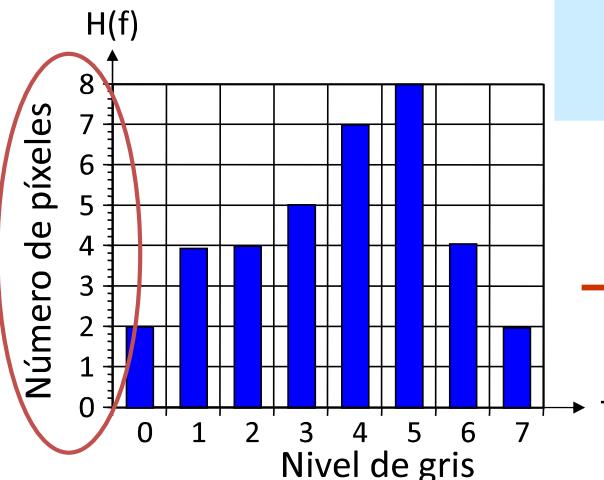
2	4	5	4	5	5
1	6	3	6	3	5
0	7	5	7	0	4
5	6	3	7	1	2
3	4	3	5	5	4
4	2	4	2	4	3

Codificada con 3 bits:
8 niveles de gris (0, 1, ..., 7)



Nivel de gris	Nº de píxeles	Frecuencia relativa
0	2	2/36
1	2	2/36
2	4	4/36
3	6	6/36
4	8	8/36
5	8	8/36
6	3	3/36
7	3	3/36
	$\Sigma=36$	$\Sigma=1$

- Definición 1: El **histograma** de una imagen se puede definir como una función discreta que representa el **número de píxeles** en la imagen en función de los **niveles de intensidad**.

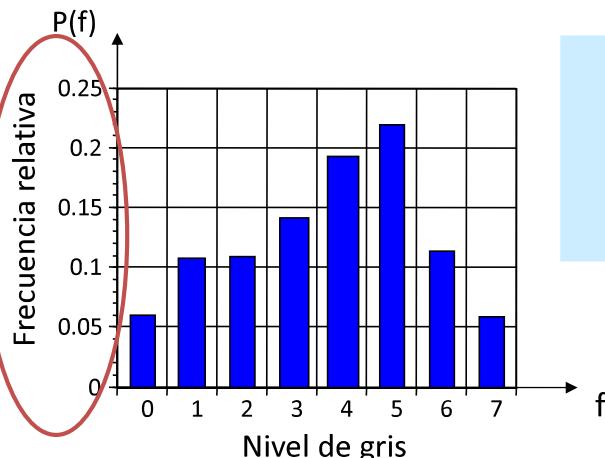


Para una imagen de dimensiones $M \times N$
y niveles de gris (intensidad) en el rango f_0 a f_k

$$\sum_{i=0}^{i=k} H(f_i) = M \times N$$

Nº de píxeles con intensidad 0: 2
Nº de píxeles con intensidad 1: 4
Nº de píxeles con intensidad 2: 4
Nº de píxeles con intensidad 3: 5
Nº de píxeles con intensidad 4: 7
Nº de píxeles con intensidad 5: 8
Nº de píxeles con intensidad 6: 4
Nº de píxeles con intensidad 7: 2

- Definición 2: El **histograma** de una imagen es la función discreta de la frecuencia relativa de ocurrencia de los píxeles de una imagen en función de los niveles de intensidad.
- La **frecuencia relativa** del histograma se puede interpretar como una **función densidad de probabilidad**.

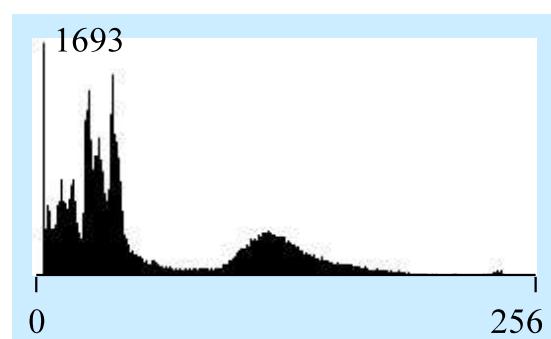
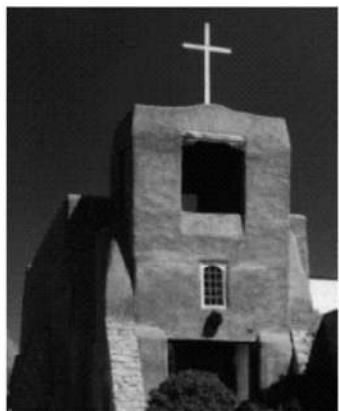
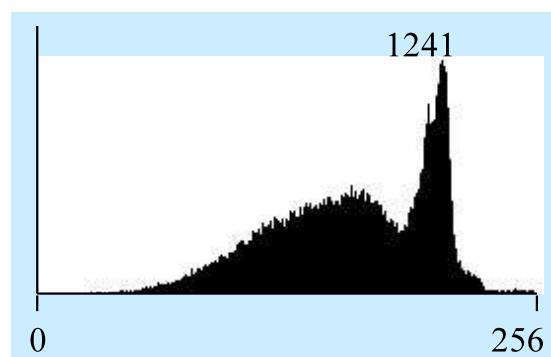


$$P(f) = \frac{H(f)}{M \times N}$$

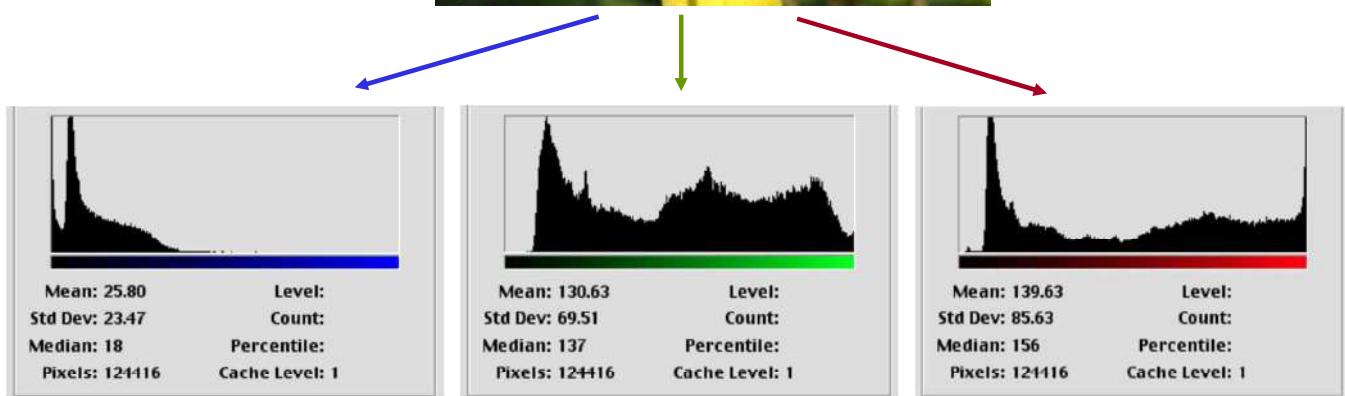
$H(f)$: Nº de píxeles con intensidad f
 $M \times N$: Tamaño imagen
 $\sum_{i=0}^k P(f_i) = 1$

La función $P(f)$ se conoce
como **función de densidad de probabilidad**

Histograma de una imagen *Ejemplos*



Histograma de una imagen *Ejemplo de imagen en color*



- **Media:** Es el valor medio de los niveles de gris. Aporta información sobre el brillo de una imagen.

$$\bar{f} = \sum_{f=0}^{L-1} f \times P(f) = \frac{1}{M \times N} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} f(i, j), \quad L = \text{número total de niveles de gris}$$

- **Varianza:** Mide la dispersión de los alrededores de la media (da idea del contraste).

$$\sigma^2 = \sum_{f=0}^{L-1} (f - \bar{f})^2 P(f)$$

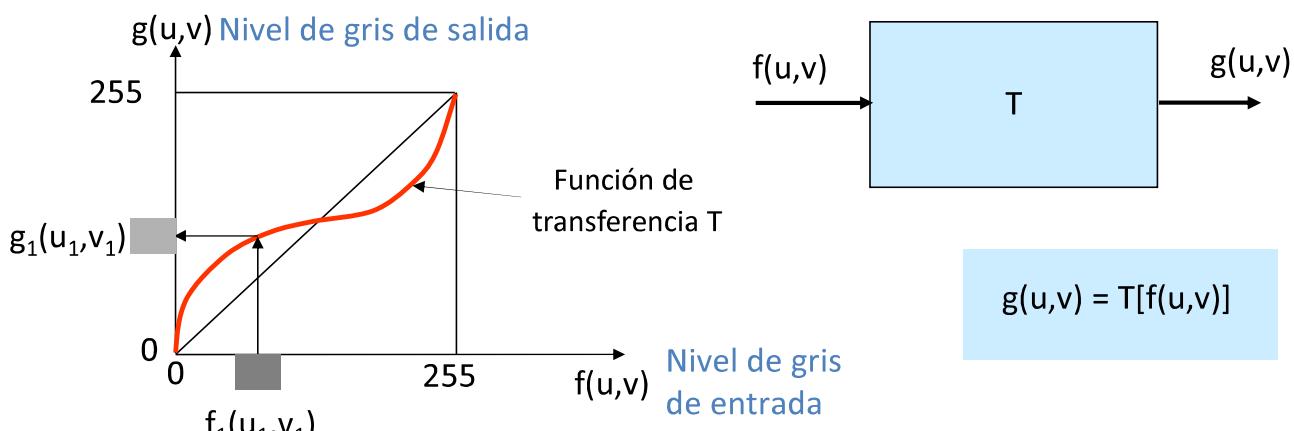
- **Asimetría sobre la media** en la distribución de los niveles de gris:

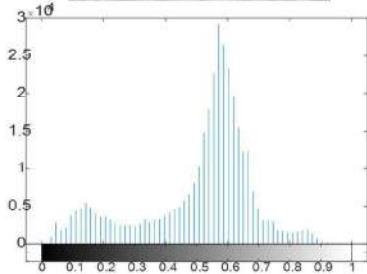
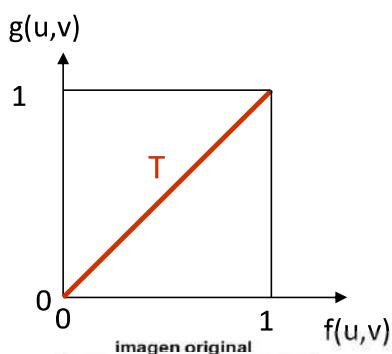
$$a = \sum_{f=0}^{L-1} (f - \bar{f})^3 P(f)$$

- **Entropía:** Informa sobre la distribución de los niveles de gris. Es una medida de su contenido de información. Una imagen con información de alta entropía, contiene mucha aleatoriedad/variabilidad y baja redundancia. Si la entropía es baja, la información de las imágenes es más predecible, conteniendo una aleatoriedad pequeña y una redundancia alta.

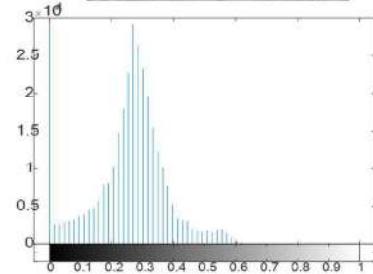
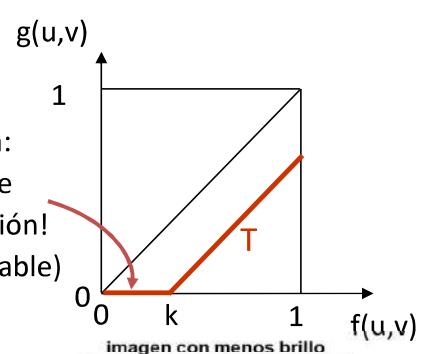
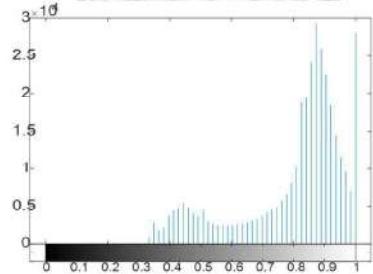
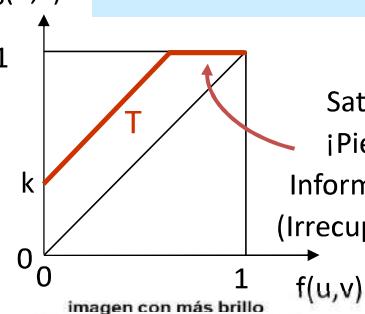
$$E = - \sum_{f=0}^{L-1} P(f) \log_2 [P(f)]$$

- Las **transformaciones del histograma** se aplican a **cada pixel** de la imagen de entrada para obtener el nivel de gris de dicho punto en la imagen de salida. En ellas, los píxeles de la imagen **se mantienen en la misma posición** (al contrario que en las transformaciones geométricas), pero **cambian de color**.
- **Funciones de transformación puntuales:** modifican **el nivel de gris** de un pixel considerando solamente su nivel de gris en la imagen original: **todos los píxeles que tengan el mismo nivel de gris en la imagen de entrada tendrán el mismo (nuevo) nivel de gris en la imagen de salida.** **No** puede ocurrir que haya píxeles en la imagen de entrada que tengan el mismo color, y en la de salida tengan uno diferente.

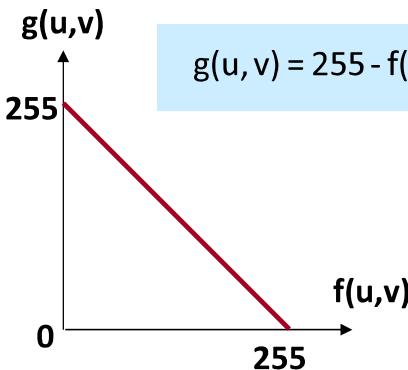




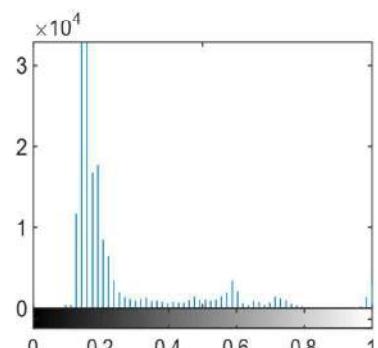
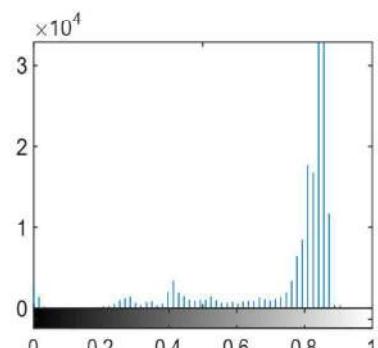
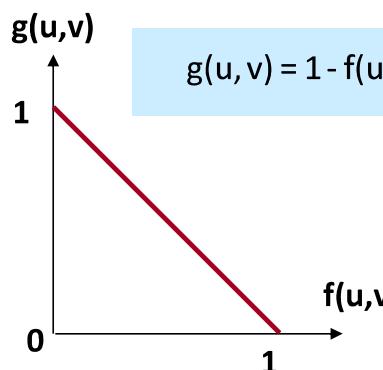
$$g(u, v) = k + f(u, v)$$

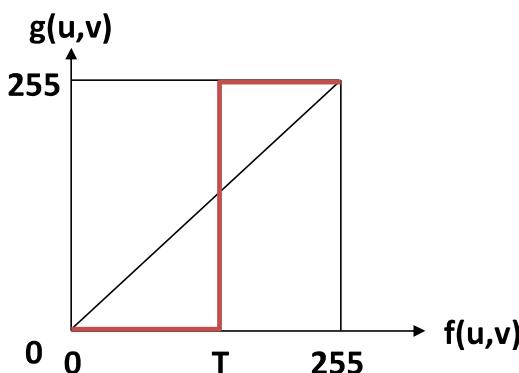


79



$g(u, v)$





$$g(u, v) = \begin{cases} 0 & \text{si } f(u, v) \leq T \\ 255 & \text{si } f(u, v) > T \end{cases}$$

T=Umbral

Imagen original



Imagen en niveles de gris



Imagen umbralizada

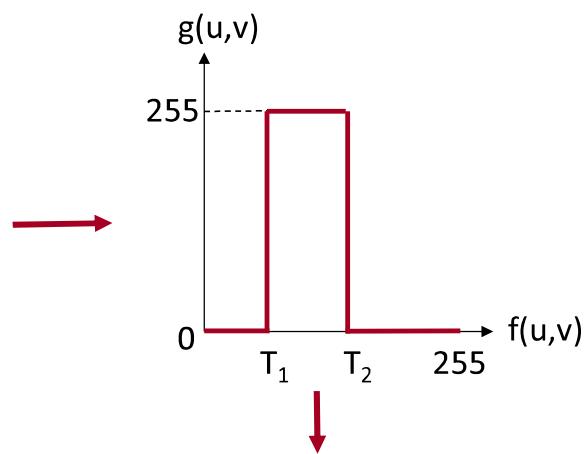


$f(u,v)$

$g(u,v)$ para $T=89$

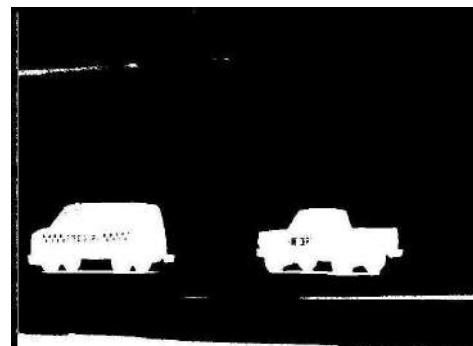


$f(u,v)$



$$g(u, v) = \begin{cases} 255 & \text{si } T_1 < f(u, v) < T_2 \\ 0 & \text{resto de casos} \end{cases}$$

$g(u,v)$



❑ Expansión de una región del histograma:

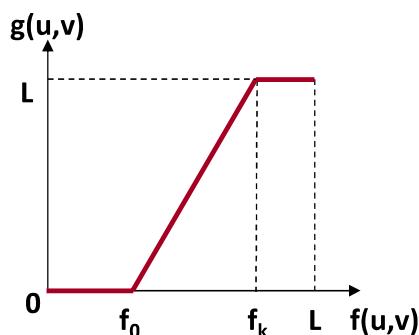
❑ Rango de variación del nivel de gris de la imagen de entrada: $[f_0, f_k]$.

❑ Variación de los niveles de gris de salida: $[0, L]$.

❑ Este algoritmo:

❑ **Expande la zona de interés:** envía los distintos niveles de gris de entrada a niveles más separados de salida (pero manteniendo todos los píxeles que tenían el mismo color en la imagen original, con el mismo (nuevo) color a la salida).

❑ **“Satura” los niveles externos a la zona de interés a 0 o 255:** todos los píxeles que en la imagen de entrada tengan nivel de gris superior a f_k tendrán a la salida el valor de 255 y los inferiores a f_0 a la salida valdrán 0. La información en esos rangos se pierde de forma irreversible.

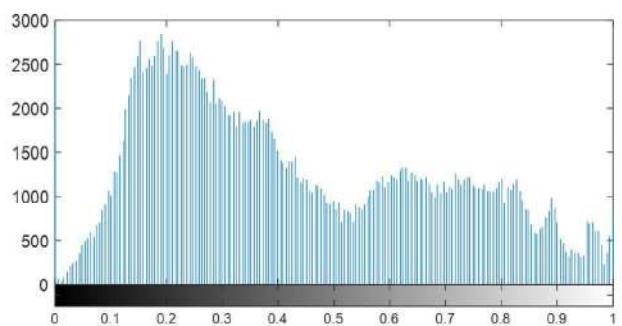
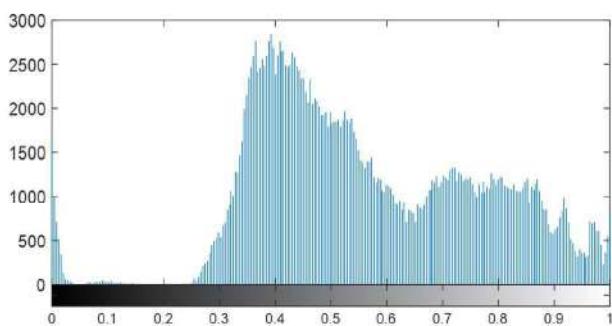


$$g(u, v) = \begin{cases} 0 & \text{si } f(u, v) < f_0 \\ \frac{f(u, v) - f_0}{f_k - f_0} L & \text{si } f_0 < f(u, v) < f_k \\ 255 & \text{si } f(u, v) > f_k \end{cases}$$

Imagen original

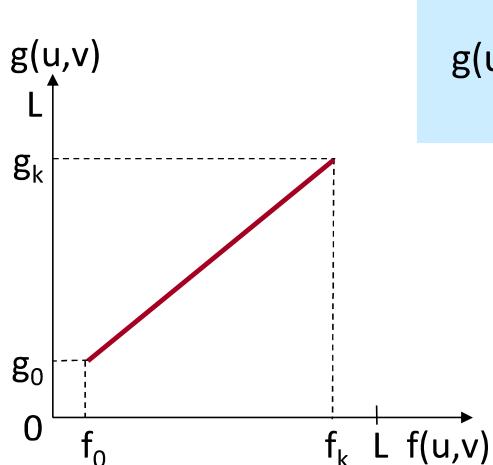


Histograma expandido



□ Contracción del histograma:

- Rango de variación del nivel de gris de la imagen de entrada: $[f_0, f_k]$. Suponemos que no hay píxeles con niveles de gris fuera de este rango, o al menos no hay píxeles que nos interesen (si los hubiera, tendríamos que definir su transformación).
- Rango deseado de la variación del nivel de gris de la imagen de salida: $[g_0, g_k]$.
- L : número total de niveles de gris posibles de nuestro sistema de representación.
- Ecuación de una recta entre los puntos $[f_0, g_0]$ y $[f_k, g_k]$:



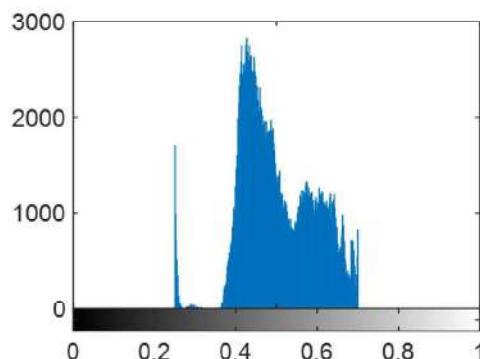
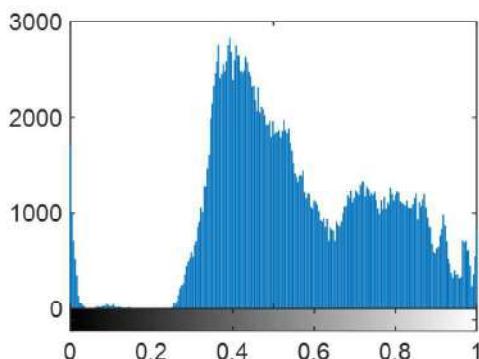
$$g(u, v) = \frac{g_k - g_0}{f_k - f_0} [f(u, v) - f_0] + g_0 = m \cdot f(u, v) + n$$

- Al contraer el histograma, habrá niveles de gris que estaban separados en la imagen original, que en la imagen de salida quedarán juntos.
- Esa acumulación puede provocar diferencias en la forma del histograma y es **irreversible**: esos niveles de gris diferentes que eran diferentes en la imagen de entrada no se podrán separar de nuevo a partir de la imagen de salida.

Imagen original

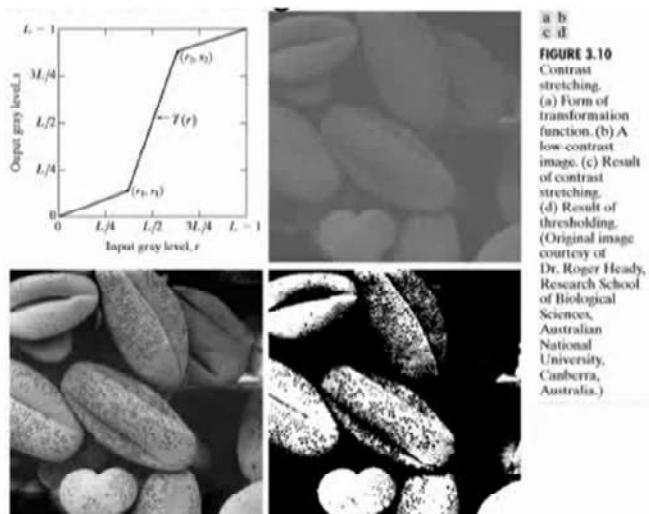


Histograma contraído

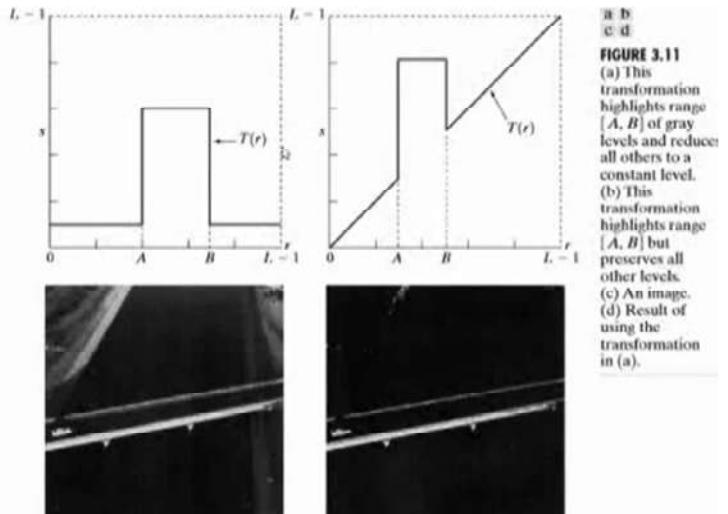


□ Funciones de transformación lineales a trozos (Piecewise-Linear)

Expansión de histograma



Gray-level slicing

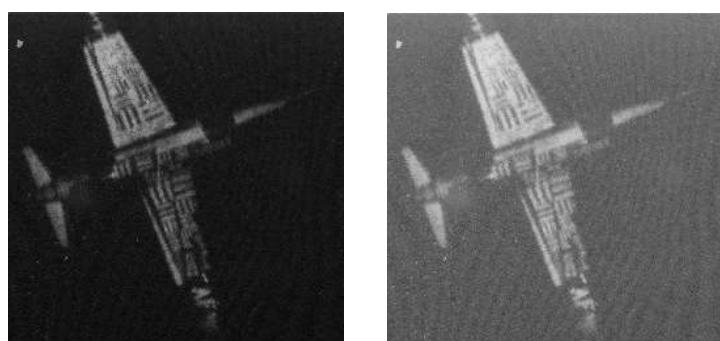
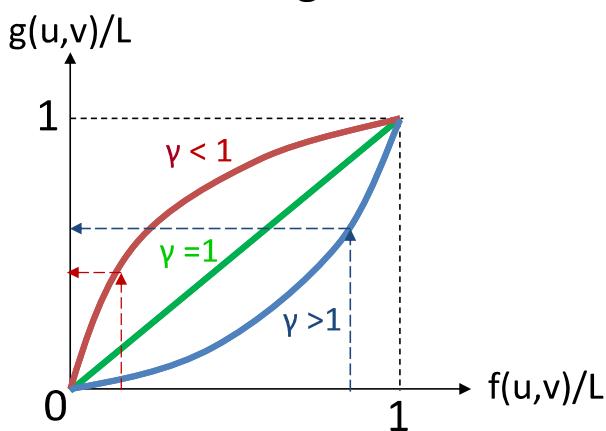


Ref.: Digital Image Processing, Second Edition By Gonzalez and Woods

28

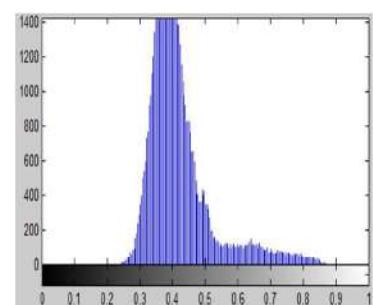
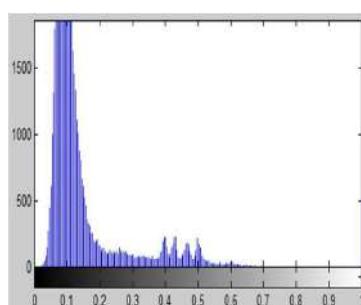
https://www.youtube.com/watch?v=O4FflvF7_SO

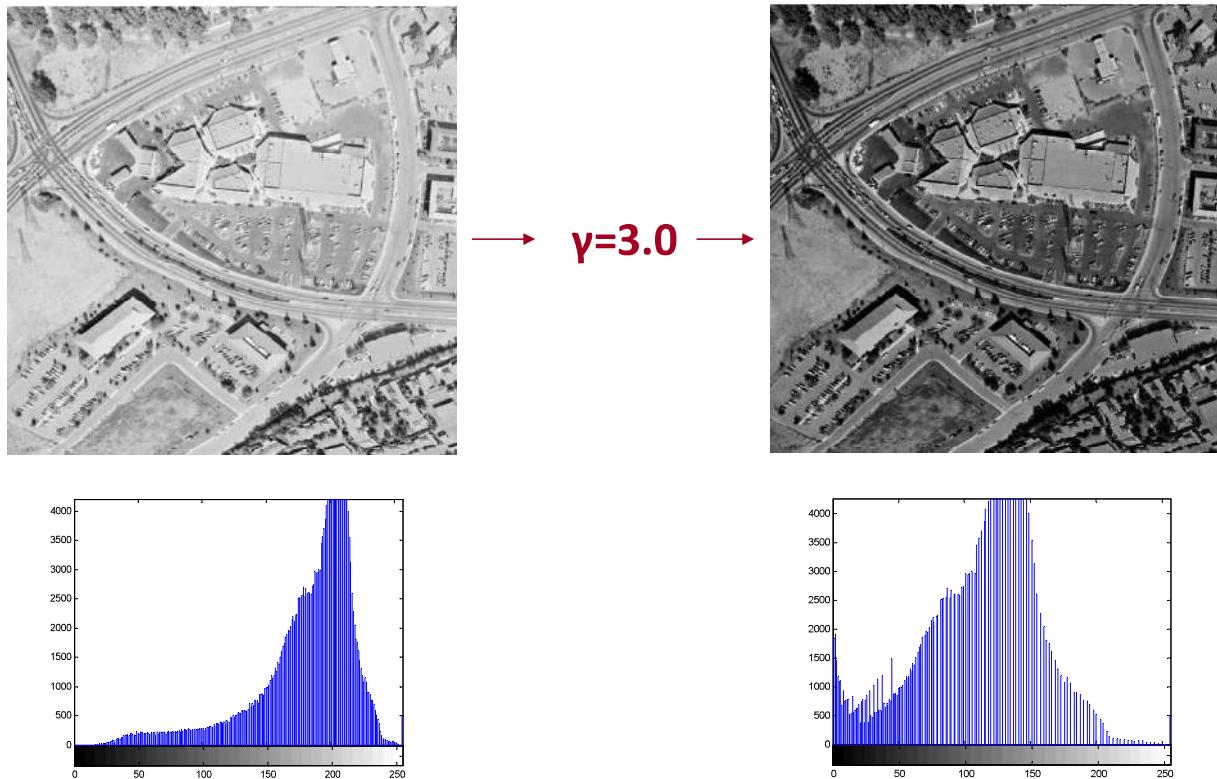
□ Corrección gamma



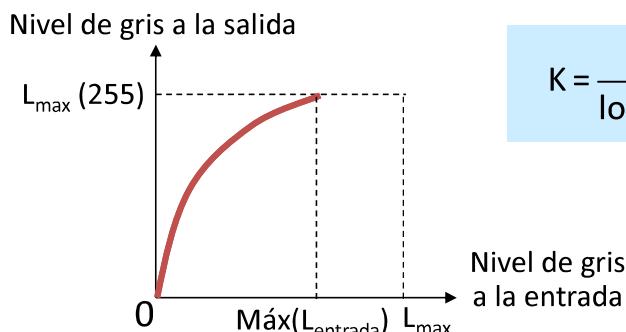
$$g(u,v) = [f(u,v)]^\gamma$$

- $\gamma < 1$: Aumenta el contraste en zonas oscuras (cerca de 0).
- $\gamma > 1$: Aumenta el contraste en zonas claras (cerca de 1).



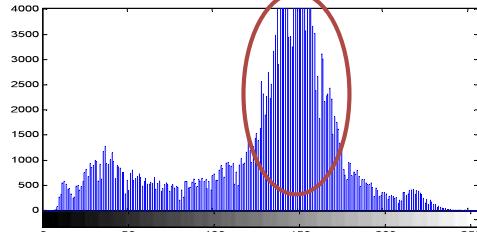


- La aplicación del **logaritmo** a los niveles de gris de una imagen tiene como resultado un **realce del contraste en la zona de niveles bajos de gris**, o zonas oscuras, a expensas de una **disminución de contraste en los niveles altos**.
- La fórmula general es:
$$g(x, y) = K * \log_{10} [1 + f(x, y)]$$
- El **1** sumado a $f(x, y)$ asegura que el nivel de gris mínimo a la salida sea el **0**.
- **K** es la constante que fija el **nivel máximo de salida**, asegurando que este no supere del máximo del sistema de representación utilizado. Se calcula de manera que al aplicar la transformación al máximo nivel de gris de la entrada, $\max\{L_{\text{entrada}}\}$, se obtenga el mayor nivel de gris soportado por el sistema de representación $L_{\text{máx}}$ (p.ej., 255):



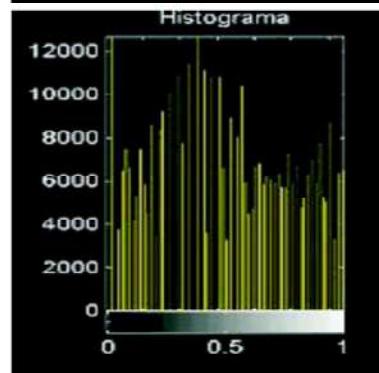
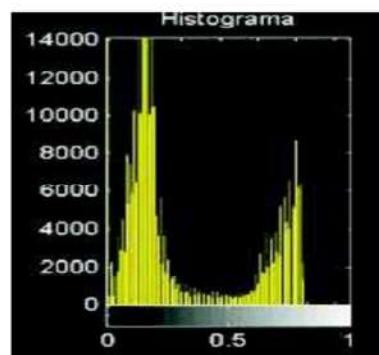
$$K = \frac{L_{\text{máx}}}{\log_{10}(1 + \max(L_{\text{entrada}}))} = \frac{255}{\log_{10}(1 + \max(L_{\text{entrada}}))}$$

- Si utilizamos un valor de **K** mayor, aumentará el contraste de los niveles bajos, a costa de saturar para los niveles más altos de gris.

- El **histograma** de una imagen tiene **picos, valles y zonas planas** bajas.
 - Picos/valles = muchos/pocos píxeles concentrados en unos pocos niveles de gris.
 - Zonas planas = un número pequeño de píxeles distribuidos sobre un amplio rango de niveles de gris.
- 

- Las **transformaciones globales del histograma** cambian cada nivel de gris de manera diferente dependiendo de la **forma** del histograma. En la fórmula de transformación se tiene en cuenta el **nivel de gris** de entrada y la forma del histograma (la **probabilidad** de todos los niveles de gris) pudiendo ocurrir que unos niveles de gris cambien mucho y otros apenas cambien de valor.
 - La aplicación de una transformación global a **distintas imágenes** dará lugar a **cambios diferentes** en los niveles de gris.
 - Todos los píxeles con el mismo nivel de gris a la entrada tendrán el **mismo** (nuevo) nivel de gris a la salida.
 - Píxeles con distinto nivel de gris a la entrada pueden acabar con el mismo nivel de gris a la salida, **perdiéndose** de forma irreversible esa información.

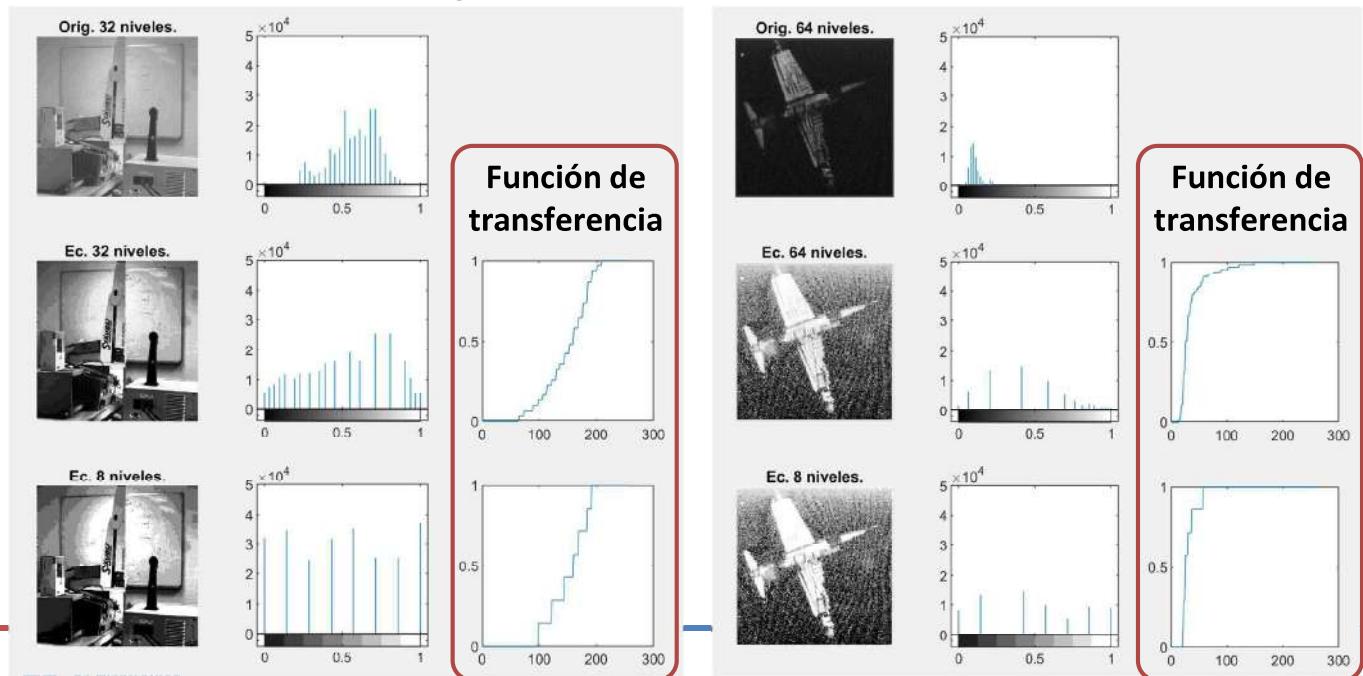
- La **ecuallización uniforme** es una de las técnicas más utilizadas para la mejora del contraste de una imagen.
- El objetivo es **modificar los niveles de una imagen de forma que el histograma de la imagen resultante sea “plano”**.
 - Separando los píxeles en los picos sobre un amplio rango de niveles de gris.
 - Acumulando las zonas planas de píxeles en rangos estrechos de niveles de gris.
- Cuando se ecualiza (como en todas las operaciones sobre el histograma), todos los píxeles con el **mismo nivel de gris en la imagen de entrada se moverán al mismo nivel de gris en la imagen de salida**, es decir, **no** puede ocurrir que dos píxeles que tengan el mismo nivel de gris en la imagen de entrada acaben en uno distinto en la imagen de salida.
- Lo que **sí** puede ocurrir es que 2 niveles próximos en la imagen de entrada acaben en el mismo nivel en la de salida.





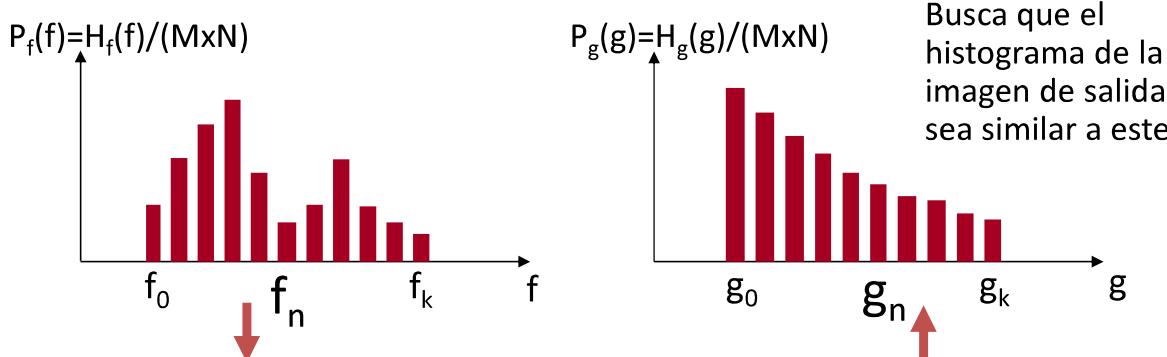
Ejemplo de: <http://dea.unsj.edu.ar/imagenes/recursos/capitulo3.pdf>

- La **función de transferencia (T)** determina en qué nivel de gris de salida se transforma cada nivel de entrada $g(u,v) = T[f(u,v)]$. Se calcula tratando el histograma como una función de densidad de probabilidad y cambiando los niveles de gris para que se asemejen a la función deseada (uniforme, exponencial, etc.).
- La **ecualización uniforme** de diferentes imágenes, con histogramas distintos, genera cambios diferentes en los niveles de gris (distintas **funciones de transferencia**).



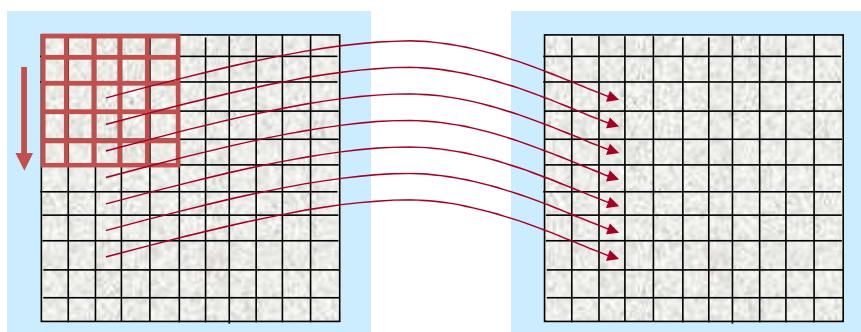
- Con un proceso similar al seguido para el caso uniforme, se puede ajustar el histograma a otras formas, como, por ejemplo, la **exponencial** o la **Rayleigh**.
- En el caso de la **ecualización exponencial**, lo que se busca es que el histograma de la imagen de salida se parezca lo más posible a:

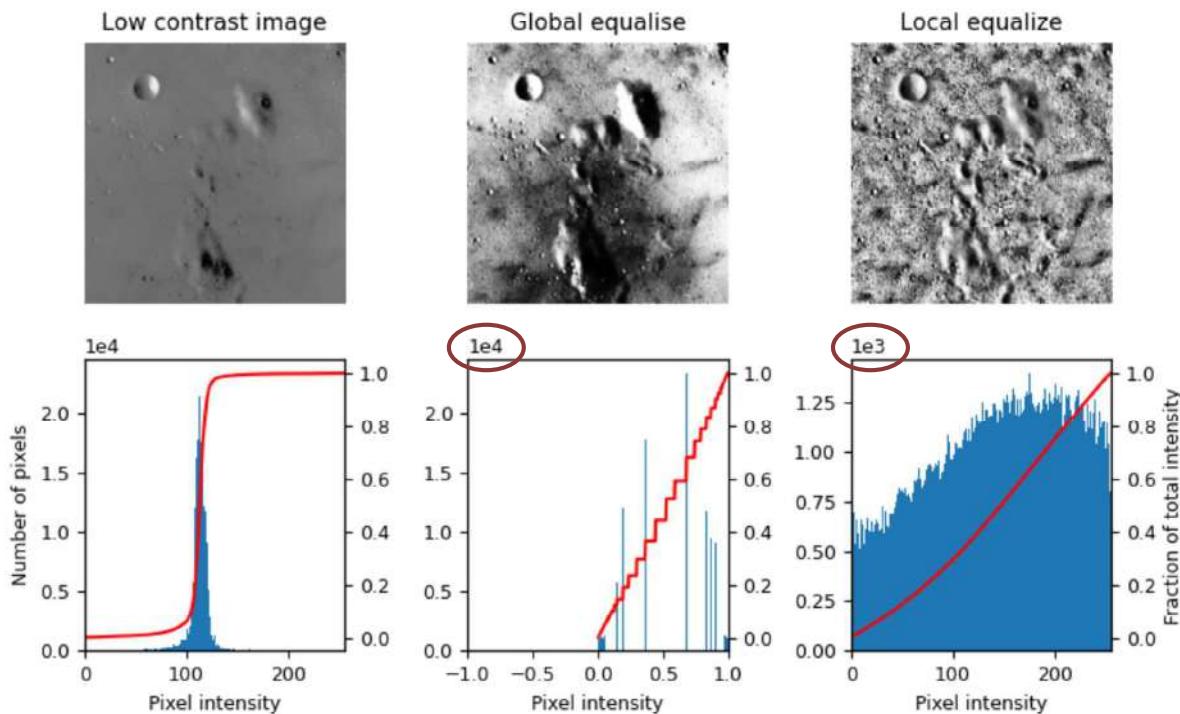
$$p_g(g) = \gamma \exp[-\gamma(g - g_0)] \quad (0 \leq \gamma \leq 1)$$



La función de transferencia asignará un valor g_n en la imagen de salida $g(u,v)$ a todos los píxeles con valor f_n en la imagen de entrada $f(u,v)$.

- Hasta ahora se ha hablado de **operaciones sobre el histograma global de una imagen**: “los píxeles se modifican mediante una función de transformación que se basa en la distribución de intensidad sobre TODA la imagen”.
- En casos prácticos, los histogramas globales **no** suelen dar buenos resultados.
- Es más frecuente realizar **operaciones locales**: Cada píxel se modifica en función de los píxeles de su entorno (modificaciones por ventanas).
- Para cada píxel en la imagen original se toma una ventana a su alrededor, se realizan las **operaciones que procedan con esa ventana** (ecualización del histograma de la ventana, etc.), y el **valor que resulte para el píxel bajo consideración será el que se le asigne en la imagen de salida**.





- Todas las transformaciones puntuales se pueden aplicar a **imágenes en color**:
 - La misma función para todas las bandas de color.
 - Diferentes funciones para las diferentes bandas de color.



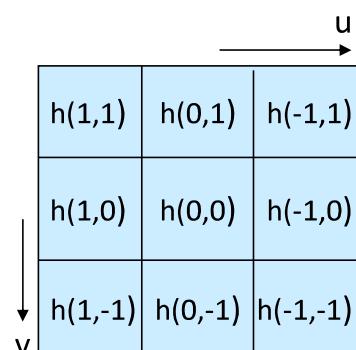
<http://www.eyemaginary.com//Rendering/ColorHistogramSpecification.pdf>

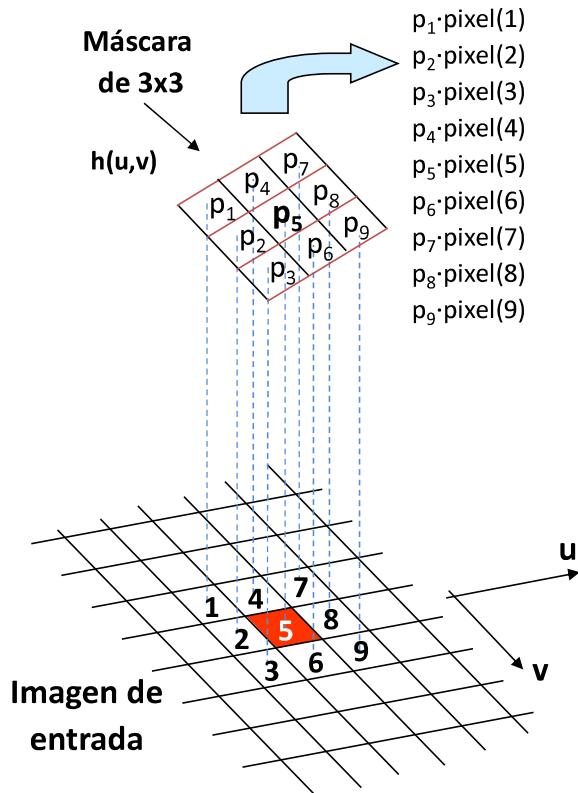
Transformadas espaciales de la imagen

Convolución espacial

□ Máscaras:

- Matrices reducidas (3×3 componentes, frecuentemente), números impares (lo que hace que exista un píxel central).
- Tamaños típicos de máscaras: 3×3 , 5×5 , 7×7 , 9×9 , 11×11 .
- El **valor** que toma cada una de las componentes de la máscara, depende de la **función** a realizar (filtro paso bajo, filtro paso alto, etc.).
- En la figura se muestra una máscara de 3×3 con valores: $h(1,1)$, $h(0,1)$, $h(-1,-1)$, $h(1,0)$, $h(0,0)$, $h(-1,0)$, $h(-1,1)$, $h(0, -1)$, $h(-1,-1)$.

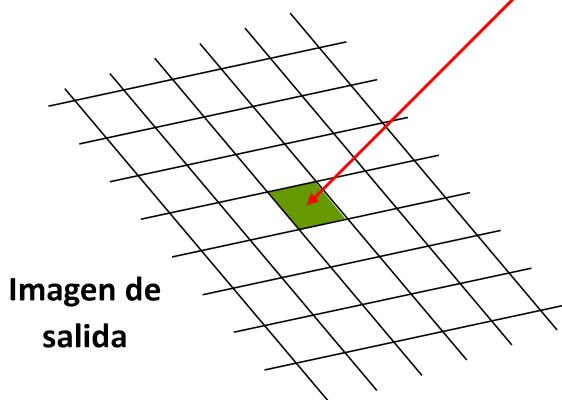




$$\begin{aligned} p_1 &= h(1,1), & p_4 &= h(0,1), & p_7 &= h(-1,1) \\ p_2 &= h(1,0), & p_5 &= h(0,0), & p_8 &= h(-1,0) \\ p_3 &= h(1,-1), & p_6 &= h(0,-1), & p_9 &= h(-1,-1) \end{aligned}$$

El proceso se repite para todos los puntos de la imagen

$$\frac{\sum_{i=1}^9 p_i \times \text{pixel}_{\text{entrada}}(i)}{\sum_{i=1}^9 p_i} = \text{valor pixel}(5) \text{ en la imagen de salida}$$



Ejemplo de convolución *Filtro paso bajo*

$$h(u,v) = \begin{pmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{pmatrix} = \frac{1}{9} \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

$f(u,v)$

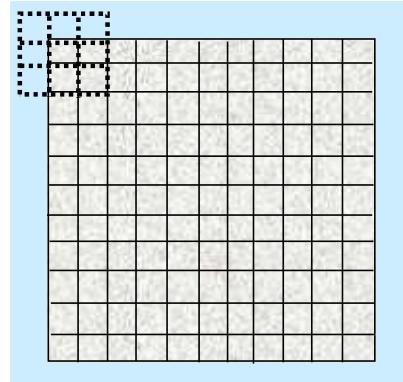
$$g(u,v) = h(u,v) * f(u,v) = \frac{1}{S_{m \times n}} \sum_i \sum_j f(i,j)h(u-i, v-j)$$

$g(u,v)$

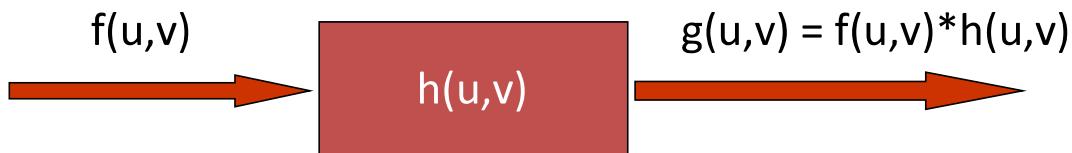


- ❑ ¿Qué sucede con **los puntos de borde** de la imagen? Hay varias soluciones:

1. Pasar la máscara por todos los píxeles de la imagen, **excepto por los de los bordes**. Esto hace que la **imagen de salida sea de menor tamaño**: Si el tamaño de imagen original es $M \times N$ y el tamaño de la máscara es $n \times n$, el tamaño de la imagen de salida es: $[M - (n - 1)] \times [N - (n - 1)]$.
2. **Añadiendo filas y columnas** en los bordes de la imagen con píxeles del **mismo valor** de intensidad que los de las filas y columnas de borde. Si la máscara es de $n \times n$ el número de filas y columnas a añadir será $(n - 1)$.
3. **Rellenar con ceros** las filas y columnas fuera de la imagen.



- ❑ La solución más frecuente es la 1.



$$g(u,v) = h(u,v) * f(u,v) = \sum_i \sum_j f(i,j)h(u-i, v-j)$$

- ❑ La **respuesta al impulso** $h(u,v)$, se suele aproximar por funciones reducidas (3x3 componentes, frecuentemente) que se suelen denominar "**máscaras**".
- ❑ Tamaño de imagen = $M \times N$
- ❑ Tamaño de máscara = $m \times n$
- ❑ Tamaño de la convolución = $[M - m + 1] \times [N - n + 1]$

$$S_{m \times n} = \sum_{i=1}^m \sum_{j=1}^n h(i,j)$$

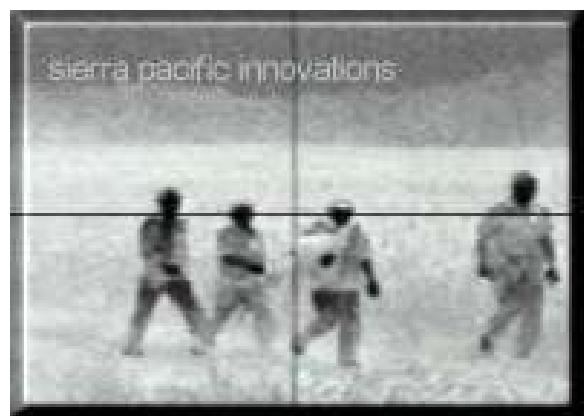
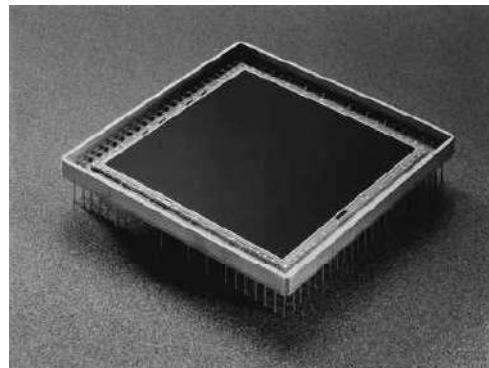
↓

$$g(u,v) = \frac{1}{S_{m \times n}} \sum_{i=-1}^1 \sum_{j=-1}^1 [f(i,j) \times h(u-i, v-j)] = f(u,v) * h(u,v)$$

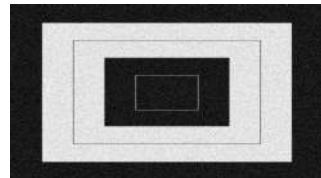
Ruido

Ruido *Origen*

- **Ruido:**
 - Cualquier variación aleatoria del valor de los píxeles de una imagen con respecto a la escena, producido en el proceso de adquisición, transmisión o procesamiento.
- **Fuentes de ruido:**
 - Sensor CCD.
 - Fluctuación de la señal en el detector.
 - Causada por energía térmica.
 - Peor en los sensores de infrarrojos.
 - Electrónica.
 - Transmisión de la imagen entre dispositivos.
 - Polvo, araños en la óptica.
 - Aberraciones ópticas de las lentes...



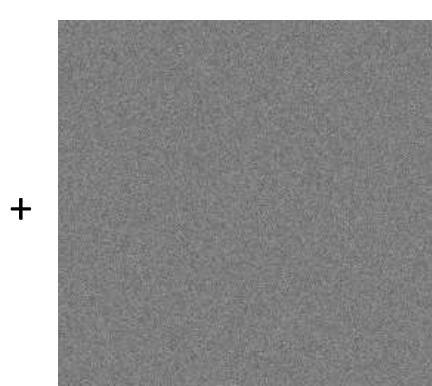
- **Ruido aditivo:** produce pequeñas variaciones en la imagen. Se **suma** a la imagen y su valor es independiente del nivel de gris del píxel: $g(x,y) = f(x,y) + n(x,y)$. Puede ser **gaussiano, exponencial, uniforme...** Tiene su origen en diferencias de ganancia del **sensor**, ruido de digitalización, perturbaciones en la transmisión, etc.
- **Ruido impulsional o ‘sal y pimienta’:** tiene un gran efecto sobre los píxeles (el ruido impone el valor del píxel) y es muy diferente a él (**blanco o negro**). Se presenta, por ejemplo, cuando se trabaja con objetos a **altas temperaturas** (problemas con infrarrojos), cuando hay polvo en la óptica, CCD defectuoso....
- **Ruido frecuencial:** La imagen es la **suma** de la imagen ideal y otra señal, la **interferencia periódica** (senoidal, etc).
- **Ruido multiplicativo:** El ruido depende de la magnitud de la señal: $g(x,y) = f(x,y) + r(x,y) \cdot f(x,y)$. La imagen obtenida es fruto de la **multiplicación** de dos señales, por ejemplo, el ruido ‘speckle’ en las imágenes de radar, sombras de objetos complejos como árboles o persianas venecianas.



- **¿Cómo se puede reducir el ruido?** Depende del tipo:
 - Promediando imágenes.
 - Realizando operaciones sobre el entorno de cada píxel.
 - Filtros lineales (paso bajo, paso alto).
 - Filtros no lineales (mediana, Gaussiano).
 - Filtros en el dominio de la frecuencia (TDF).
- **Ejemplo de ruido aditivo:**



Imagen



+

Ruido



=

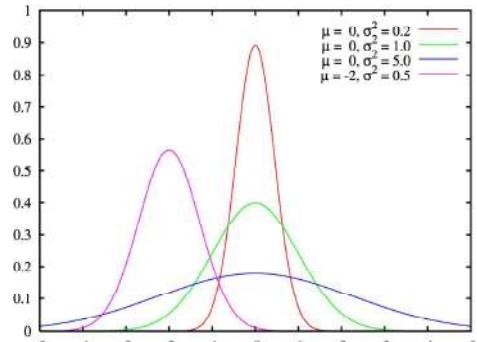
Imagen con ruido

- La distribución típica de ruido es **Gaussiana con $\mu=0$ y desviación típica: σ** .
- La fórmula de la gaussiana unidimensional es:

$$\eta(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2} \frac{(x-\mu)^2}{\sigma^2}\right)$$

- Esta distribución nos da la **probabilidad** de que el ruido (x) tenga un determinado valor, es decir, **modifique el nivel de gris de un pixel cualquiera de la imagen original** en un cierto nivel:

- El pico en la zona de la **media** significa que es más probable que el valor esté cerca de la media (en nuestro caso, que la media es 0, de que sean valores pequeños).
- Cuando la **desviación típica** es **pequeña**, será muy probable que los valores del ruido sean muy pequeños,
- si la **desviación típica** es más **elevada**, hay una probabilidad mayor de que el ruido tenga valores más grandes.



https://upload.wikimedia.org/wikipedia/commons/1/1b/Normal_distribution_pdf.png

Ruido: filtros lineales

Promediado de imágenes

- Considerando una **imagen ruidosa $f(u,v)$** , está se puede expresar como:

$$f(u,v) = f_{sin_ruido}(u,v) + n(u,v)$$

siendo $n(u,v)$ el **ruido**: el valor que ha cambiado el pixel (u,v) , que será un número pequeño, distinto para cada pixel, ya que el ruido es **incorrelado, gausiano**, de **media cero** y **varianza σ_n^2** .

- La **media** de varias imágenes, captadas en las **mismas condiciones**:

$$g(u,v) = \bar{f}(u,v) = \frac{1}{K} \sum_{i=1}^K f_i(u,v)$$

- La **varianza** de los píxeles (respecto a su valor ideal) en la imagen resultante $g(u,v)$ viene dada por:

$$\sigma_g^2(u,v) = \frac{1}{K} \sigma_n^2(u,v)$$

- Es decir, si podemos capturar **n imágenes iguales** y **hacemos la media**, la **imagen resultante tendrá menos ruido gausiano (de menor varianza)**. Esta técnica se denomina **promediado de imágenes** y el principal problema a la hora de aplicarlas es conseguir **n imágenes exactamente iguales**.

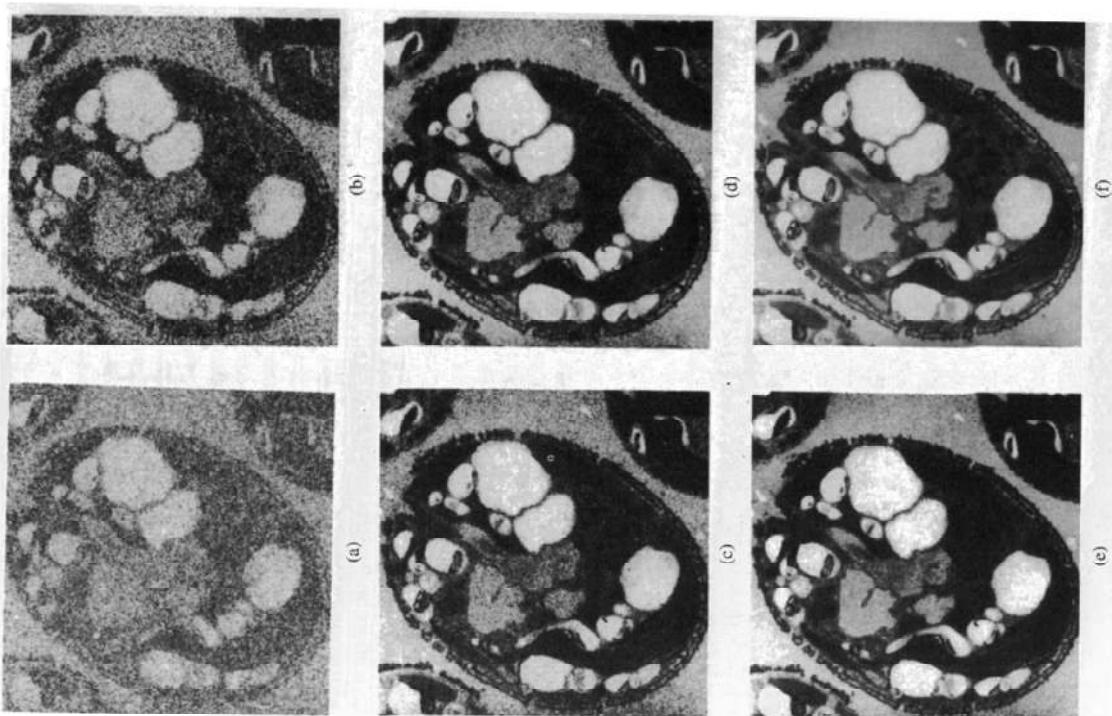


Figura 4.18. Ejemplo de reducción de ruido por promediado: (a) una imagen típica con ruido; (b)-(f) resultados de promediar 2, 8, 16, 32 y 128 imágenes con ruido.

- **Promediado del entorno de vecindad:** se calcula el valor del píxel de la imagen de salida haciendo la **media** de los píxeles del entorno en la imagen de entrada.

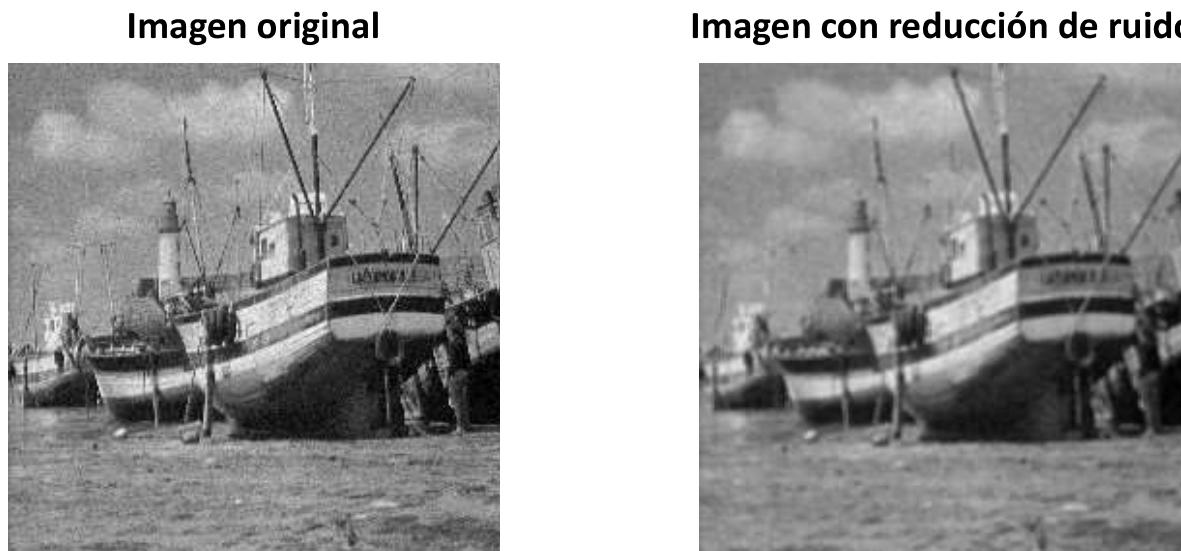
$$h(u, v) = \begin{pmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{pmatrix} = \frac{1}{9} \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

$\rightarrow g(u, v) = h(u, v) * f(u, v) = \frac{1}{S_{mn}} \sum_i \sum_j f(i, j)h(u - i, v - j)$

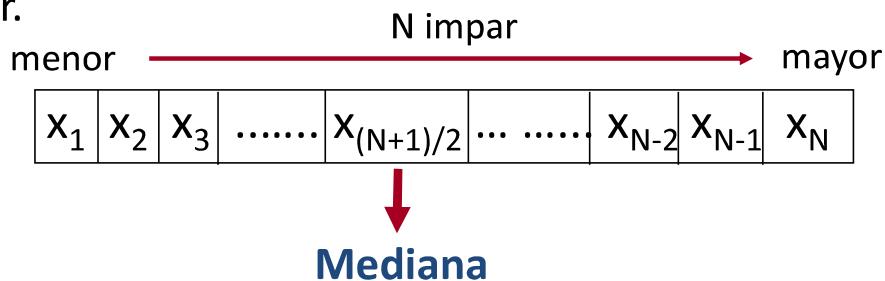
$f(u, v)$ $g(u, v)$



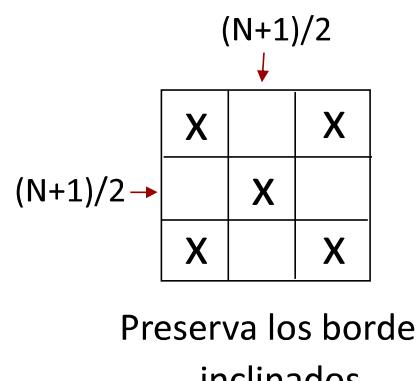
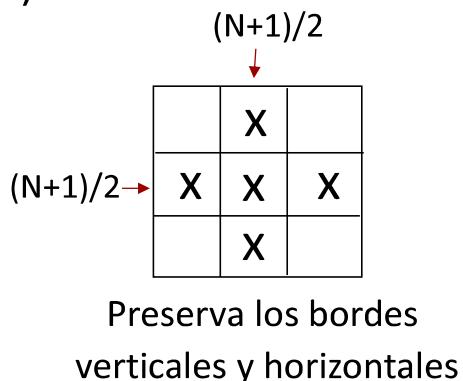
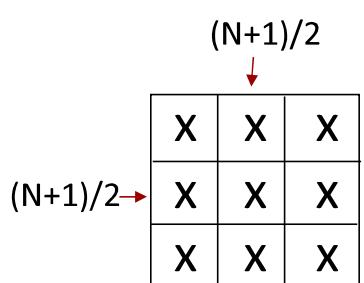
- La reducción de ruido **promediando** puede conllevar la **desaparición de detalles finos en la imagen**.



- **Unidimensional** ($N \times 1$). **Mediana**: valor central de la ordenación de menor a mayor.



- **Bidimensional** ($N \times N$)



- El filtro de mediana es no lineal y es más útil que el promediado para eliminar ruido impulsional.



Figura 4.23. (a) Imagen original; (b) imagen corrompida por ruido en forma de impulsos; (c) resultado del promedio en un entorno 5×5 ; (d) resultado producido por un filtro de mediana de tamaño 5×5 . (Cortesía de Martin Connor, Texas Instruments Inc., Lewisville, Tex.)

Original



Filtro paso bajo

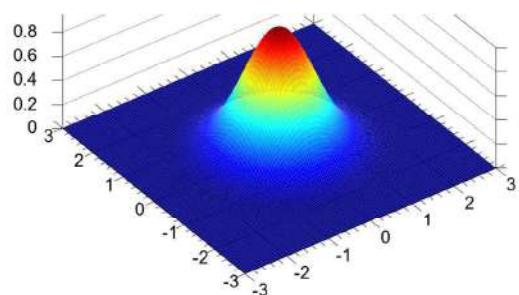


Mediana



- **Filtro Gaussiano:** Partimos de una **gaussiana bidimensional** con media 0 para ambas dimensiones y con la misma desviación típica (σ) para las dos:

$$G(u, v) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(u^2 + v^2)}{2\sigma^2}\right)$$



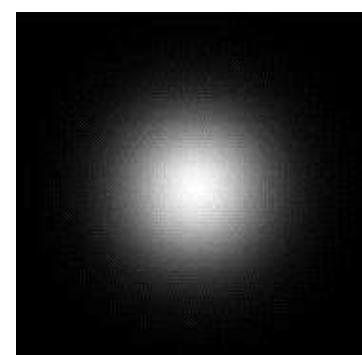
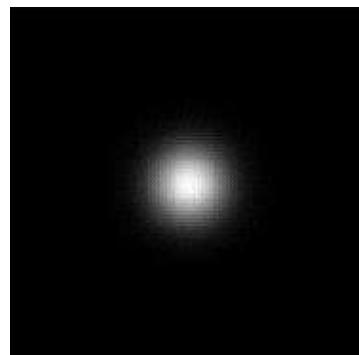
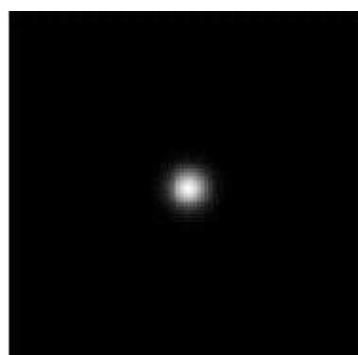
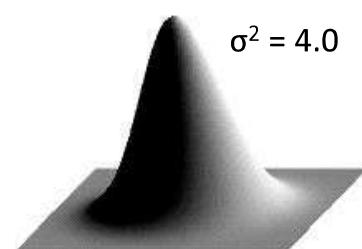
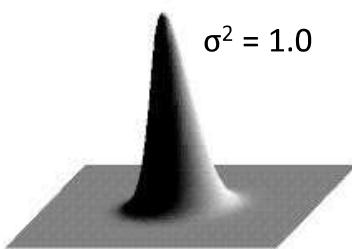
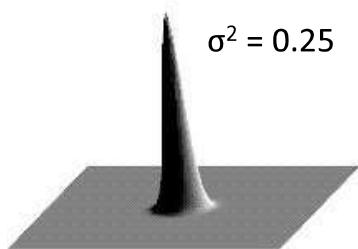
https://upload.wikimedia.org/wikipedia/commons/3/3e/Gaussian_2d.svg

- Fijado un valor de σ^2 , la función $G(u,v)$ se puede aproximar por una **máscara**, cuyas dimensiones dependen del valor de σ^2 :

G(-1,-1)	G(0,-1)	G(1,-1)
G(-1,0)	G(0,0)	G(1,0)
G(-1,1)	G(0,+1)	G(1,1)

- Se van dando pares de valores a u y v .
- El tamaño de la máscara se trunca cuando los valores de $G(u,v)$ sean despreciables frente a los otros.
- Los valores de $G(u,v)$ se pueden escalar (multiplicar por una constante) y redondear al entero más próximo.
- Se divide por la suma de todos los valores para que no incremente el brillo en la imagen de salida.

$$G(u, v) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(u^2 + v^2)}{2\sigma^2}\right) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{u^2}{2\sigma^2}\right) \cdot \exp\left(-\frac{v^2}{2\sigma^2}\right)$$



$$\sigma = 0.391$$

1	4	1
4	12	4
1	4	1

$$\frac{1}{32}$$

$$\sigma = 0.625$$

1	2	3	2	1
2	7	11	7	2
3	11	17	11	3
2	7	11	7	2
1	2	3	2	1

$$\frac{1}{121}$$



Máscara Gaussiana
7x7



Máscara Gaussiana
15x15



Visión Artificial & Sistemas de Visión Artificial

Tema 3: Operaciones Morfológicas

Autores: Sira Palazuelos, Luis M. Bergasa , Manuel Mazo,
M. Ángel García, Marisol Escudero, J. Manuel Miguel
Departamento de Electrónica. Universidad de Alcalá.

❑ Operaciones Morfológicas:

- ❑ Erosión
- ❑ Dilatación
- ❑ Closing (Cierre)
- ❑ Opening (Apertura)
- ❑ Top Hat (White Top Hat)
- ❑ Bottom Hat (Black Top Hat)
- ❑ Gradiente Morfológico

- Las **operaciones morfológicas** implican desplazar un elemento estructurante o kernel (B) por toda la imagen (A) y realizar las operaciones específicas sobre la superposición entre A y B, y asignar el resultado al píxel central (**punto de anclaje**).
- El **elemento estructurante** define el entorno de vecindad a través de una matriz de "1s" y "0s".
 - El píxel central (**punto de anclaje**) representa el píxel de interés.
 - Los elementos de la matriz que están en "1" definen el **entorno de vecindad**.
- Ejemplos de elementos estructurantes:

1	1	1
1	1	1
1	1	1

0	1	0
1	1	1
0	1	0

1	0	1
0	1	0
1	0	1

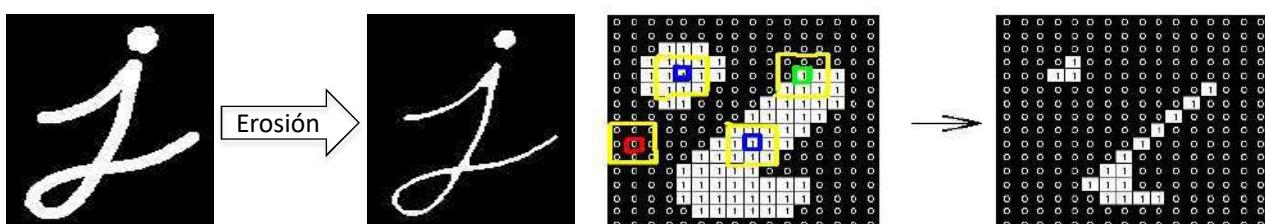
Operaciones Morfológicas Erosión

□ Erosión

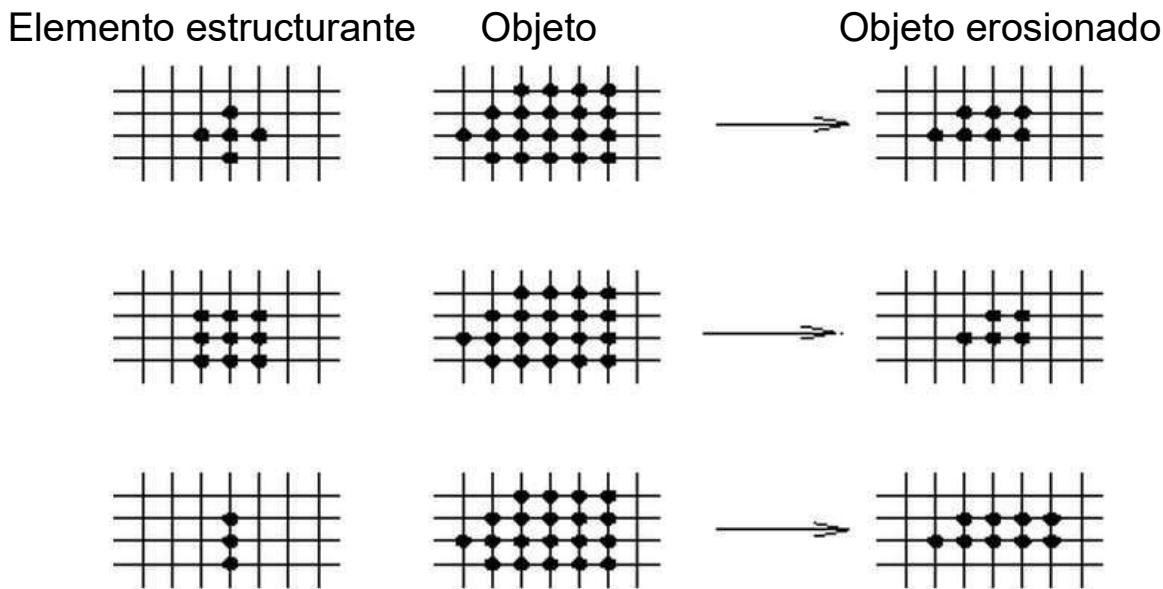
- **Regla:** Dada una imagen A, el pixel central del elemento estructurante B tomará, en la imagen de salida, el **valor menor** de A cubierto por B. Si la imagen original es en blanco y negro, la imagen de salida es:

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

- Si el primer plano es brillante sobre un fondo oscuro, entonces:
 - El **efecto** de la erosión es "**adelgazar**" los objetos.
 - **Elimina los objetos más pequeños** que el elemento estructural.
 - Si se aplica varias veces, todos los objetos desaparecen.



- Si un objeto se erosiona con **diferentes** elementos estructurales, el resultado es distinto.



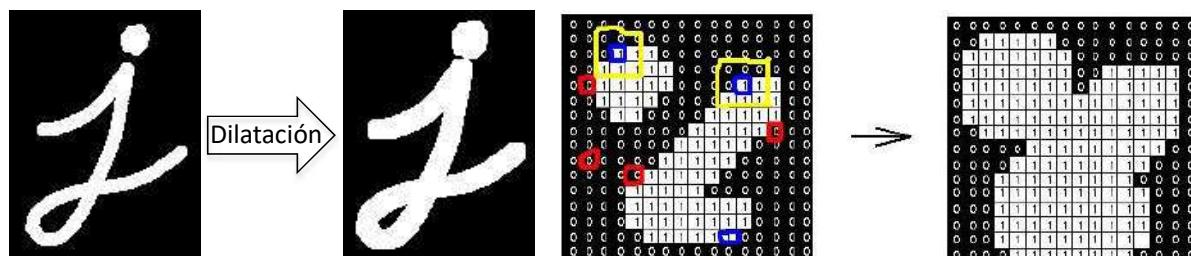
http://utam.gg.utah.edu/tomo03/03_mid/HTML/node103.html

□ Dilatación

- **Regla:** Dada una imagen A, el pixel central del elemento estructurante B tomará, en la imagen de salida, el **valor mayor** de A cubierto por B. Si la imagen original es en blanco y negro, la imagen de salida es:

$$A \oplus B = \{x | [B_x \cap A] \neq \emptyset\}$$

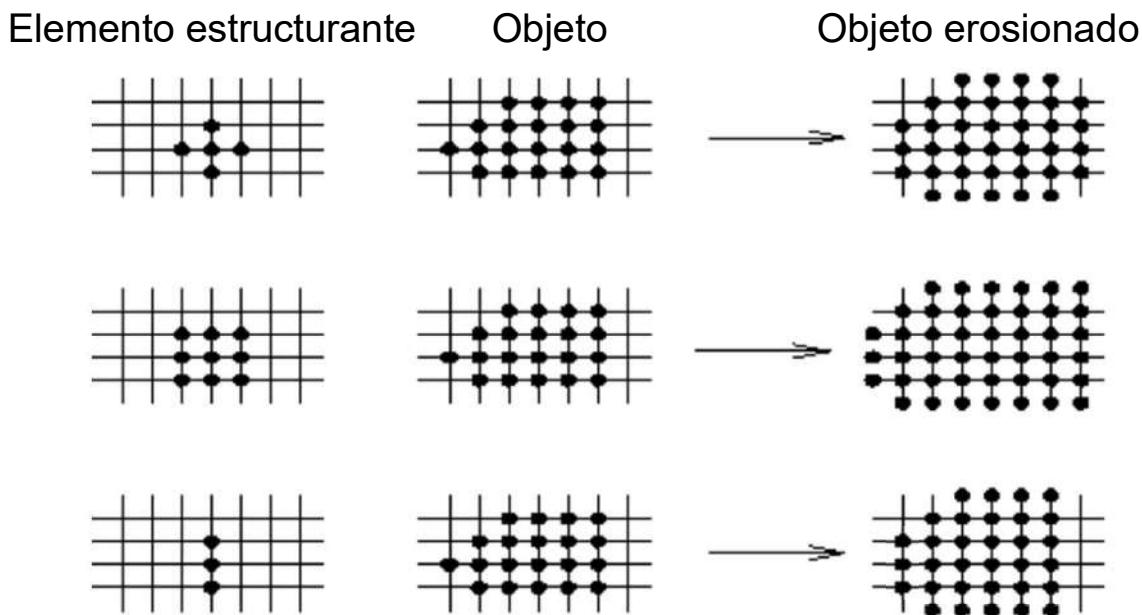
- Si el primer plano es brillante sobre un fondo oscuro, entonces:
 - El **efecto** de la dilatación es "engordar" los objetos.
 - **Elimina los agujeros más pequeños** que el elemento estructural.
 - Si se aplica varias veces, el fondo desaparece.



https://docs.opencv.org/2.4/doc/tutorials/imgproc/erosion_dilatation/erosion_dilatation.html

<http://chilmania.blogspot.com.es/2010/06/matlab-code-for-dilation-and-erosion.html>

- Si un objeto se dilata con **diferentes** elementos estructurales, el resultado es distinto.



http://utam.gg.utah.edu/tomo03/03_mid/HTML/node103.html

□ Closing (Cierre)

- Es una dilatación seguida de una erosión.

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

- Rellena/cierra los pequeños **agujeros** que existen en la imagen, si el primer plano es brillante sobre un fondo oscuro.
- Satisface la propiedad de la **idempotencia**: después de la primera aplicación, otras aplicaciones no producen ningún cambio en el resultado.



https://docs.opencv.org/3.2.0/d9/d61/tutorial_py_morphological_ops.html

□ Opening (Apertura)

- Es una erosión seguida de una dilatación.

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

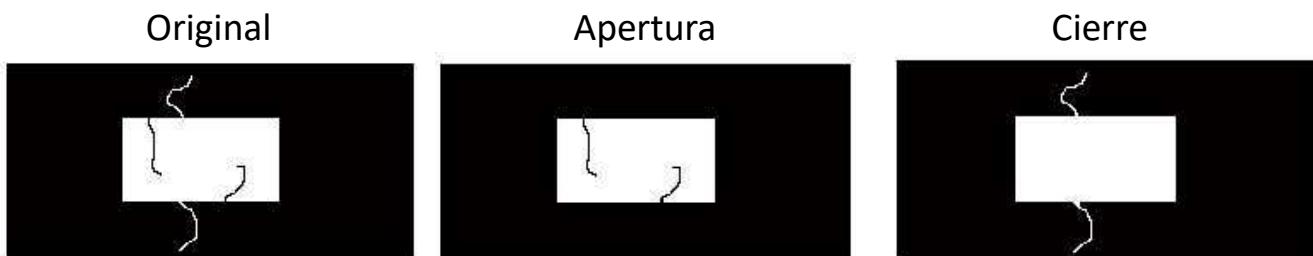
- Elimina pequeños **píxeles aislados** de la imagen, si el primer plano es brillante sobre un fondo oscuro.
- Satisface la propiedad de la **idempotencia**: después de la primera aplicación, otras aplicaciones no producen ningún cambio en el resultado.



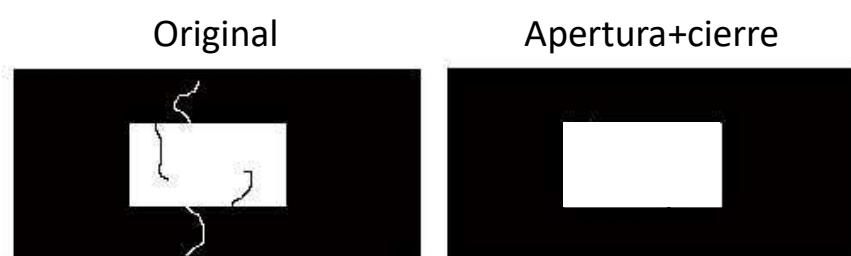
https://docs.opencv.org/3.2.0/d9/d61/tutorial_py_morphological_ops.html

Morphological operations

□ Apertura y cierre



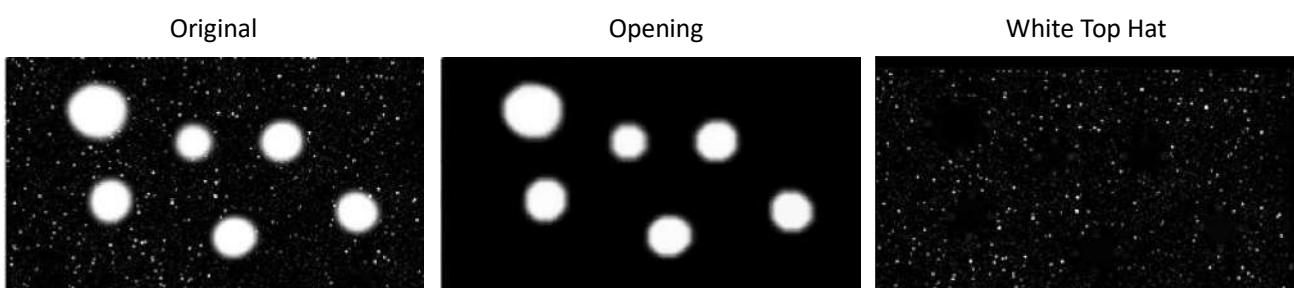
- Para deshacernos tanto de los objetos pequeños como de los agujeros pequeños tendremos que ejecutar secuencialmente apertura y cierre.



- Las transformaciones anteriores eliminan los detalles pequeños de una imagen.
- Top hat** y **bottom hat** nos permiten extraer y resaltar los elementos pequeños y detalles de las imágenes. Se utilizan para extraer características (*feature extraction*), para mejorar imágenes (*image enhancement*), etc.
- Top hat (white top hat)**
 - Es la diferencia entre la imagen original (A) y una apertura ($A \circ B$).

$$I = A - A \circ B$$

- Permite enfatizar objetos de la imagen de entrada que sean pequeños y claros sobre fondo negro:
 - Más pequeños** que el elemento estructurante, y
 - Más brillantes** que sus alrededores.



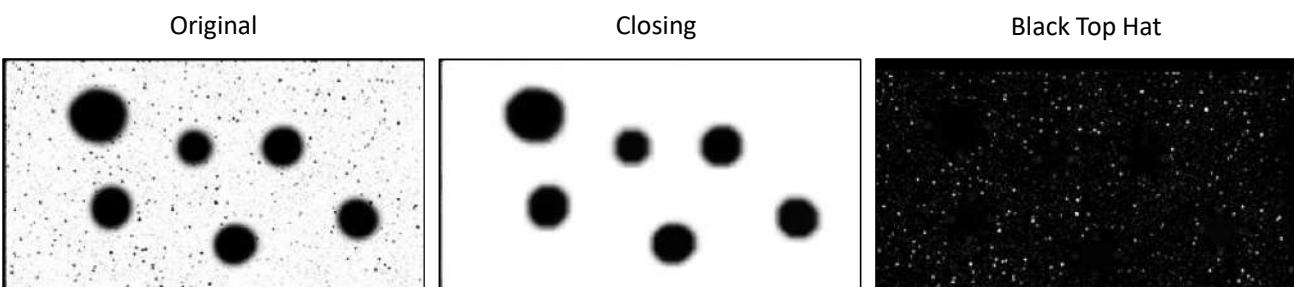
https://en.wikipedia.org/wiki/Top-hat_transform

□ Bottom hat (Black top hat, black hat)

- Es la diferencia entre un cierre ($A \bullet B$) y la imagen original (A).

$$I = (A \bullet B) - A$$

- Permite enfatizar objetos de la imagen de entrada que sean pequeños y oscuros sobre fondo blanco:
 - Más pequeños** que el elemento estructurante, y
 - Más oscuros** que sus alrededores.



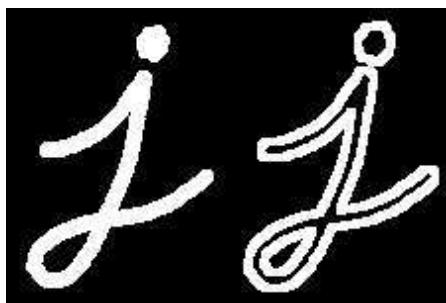
https://en.wikipedia.org/wiki/Top-hat_transform

□ Gradiente morfológico

- Es la diferencia entre la dilatación y la erosión de una imagen.

$$I = (A \oplus B) - (A \ominus B)$$

- Es útil para encontrar el **contorno** de un objeto, como se puede ver en la imagen



https://docs.opencv.org/3.2.0/d9/d61/tutorial_py_morphological_ops.html



Sistemas de Visión Artificial

Tema 3. Técnicas de Segmentación

Autores: Sira Palazuelos, Luis M. Bergasa , Manuel Mazo,
M. Ángel García, Marisol Escudero, J. Manuel Miguel

Departamento de Electrónica. Universidad de Alcalá.

Índice

- 1. Introducción**
- 2. Segmentación basada en regiones**
 - Growing regions
 - Splitting & Merging regions
 - Transformada de Watershed
- 3. Segmentación basada en el histograma**
 - Análisis Global frente a Análisis Local
 - Umbralización
 - Métodos de clustering (K-means)
 - Métodos estocásticos
- 4. Detección de discontinuidades**
 - Detección de bordes
 - Detección de esquinas
- 5. Segmentación basada en bordes**
 - Análisis local
 - Análisis global



1. Introducción

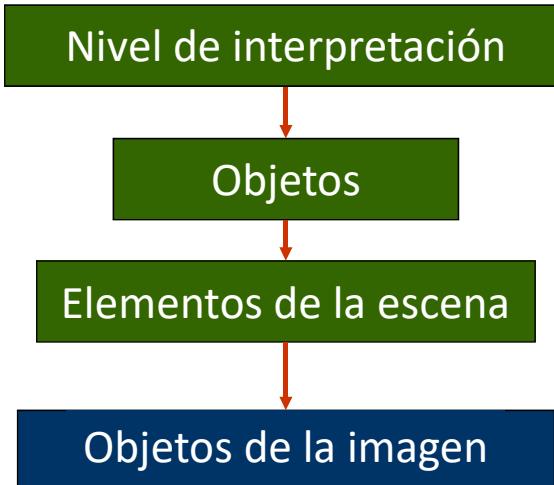
Introducción

- **Segmentar** una imagen es **dividirla** en las partes u objetos de interés que la componen.
- Los **métodos de segmentación** buscan zonas que tengan características **homogéneas** dentro de ellas y que **contrasten** con las de alrededor.
- Las medidas de homogeneidad y contraste incluyen **características** como: **nivel de gris, color, textura, bordes, movimiento,** etc.
- Estas zonas con valor del atributo uniforme se considerarán **objetos** de la imagen.



<http://www.formacionyeducacionvial.com/placas-de-matricula/>

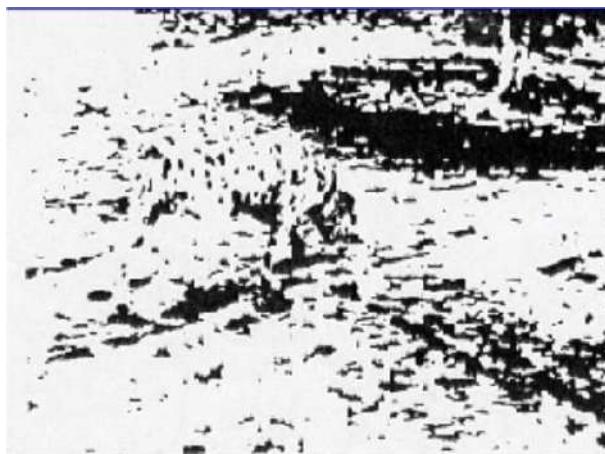
- Correspondencia entre una escena real y su imagen asociada:



- ¿Qué objetos de una imagen se corresponden con elementos con significado de una escena?
- La respuesta depende del **tipo y complejidad** de la imagen.
- El concepto de **segmentación no está bien definido**.

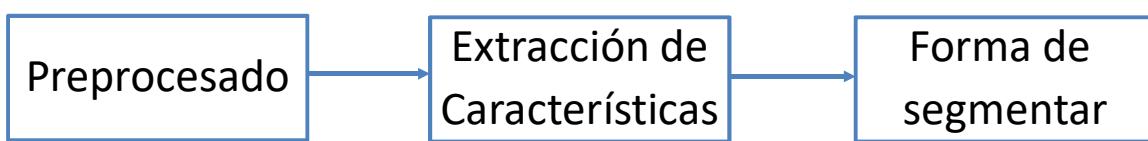


- Evaluación de la **calidad** de la segmentación:



- A veces es una tarea difícil.

- Esquema general de funcionamiento



- Eliminar lo que no interesa

- Preservar la información de contornos

- Espacio de trabajo

- Nivel de gris
- Bordes
- Color
- Texturas
- Movimiento

- Técnicas de segmentación:

- Basada en regiones
- Basada en histograma
- Basada en bordes



2. Segmentación basada en Regiones

Segmentación basada en regiones *Formulación básica*

- La **segmentación en regiones** de una imagen, $f(u,v)$ representada mediante la región R , es un proceso de partición de R en un conjunto de K regiones $\{R_j\}$, $1 \leq j \leq K$, tal que:
 - La unión de todas las regiones forma la imagen $\rightarrow R = \bigcup_{i=1}^K R_i$
 - Ningún píxel pertenece a más de una región:

$$\rightarrow R_i \cap R_j = \emptyset \quad \forall i, j ; i \neq j$$

- Coherencia espacial
 - $\rightarrow R_i$ es una región conectada
- Coherencia de características \rightarrow Para algún predicado P :

$P(R_i)$ es CIERTO para $i = 1, 2, \dots, K$

$P(R_i \cup R_j)$ es FALSO para R_i, R_j adyacentes e $i \neq j$

- **Método de crecimiento de regiones:** Para **segmentar** la imagen se comienza con un conjunto de puntos **semilla** y se van haciendo crecer las regiones a su alrededor, añadiendo a cada punto semilla aquellos píxeles vecinos que tengan **propiedades similares**.
- **Idea general:**
 - Un píxel se añade a una región si se cumplen dos condiciones:
 - El píxel debe ser **adyacente** a la región.
 - El píxel debe ser **similar** a los píxeles de esa región.
 - El proceso continúa hasta que **no** se puedan añadir más puntos.
 - Se elige **otro punto de semilla**, no perteneciente a ninguna de las regiones previas, y el proceso se repite hasta que se segmenta la totalidad de la imagen.

- **Criterio de similitud:** ¿Cuándo se considera un píxel similar para ser añadido a una región?
 - Cuando la **diferencia de características (predicados)** con respecto a los píxeles de la región es inferior a un determinado **umbral (T)**.
 - Ej: consideremos el nivel de gris:

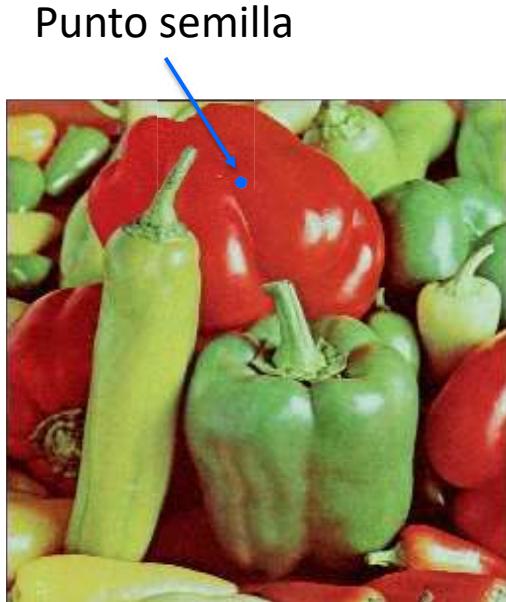
0	0	5	6	7
1	1	5	8	7
0	1	6	7	7
2	0	7	6	6
0	1	5	6	5

$$|P(p) - P(p_{\text{adyacente}})| < T$$

0	0	5	6	7
1	1	5	8	7
0	1	6	7	7
2	0	7	6	6
0	1	5	6	5

0	0	5	6	7
1	1	5	8	7
0	1	6	7	7
2	0	7	6	6
0	1	5	6	5

□ Ejemplo de crecimiento de regiones con distintos umbrales

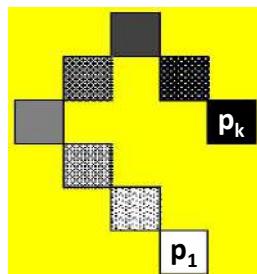


□ Problemas:

- Selección de las **semillas** iniciales (número y ubicación).
- Criterio de **similitud**:
 - **Características** a tener en cuenta (color, movimiento, etc.).
 - Fijación del **umbral**.

□ Debilidades:

- Deriva:



Secuencia de píxeles:

$$p_1, \dots, p_k$$

$$p_j, p_{j+1} \text{ entorno}$$

$$|P(p_j) - P(p_{j+1})| < T \text{ pero } |P(p_1) - P(p_k)| > T$$

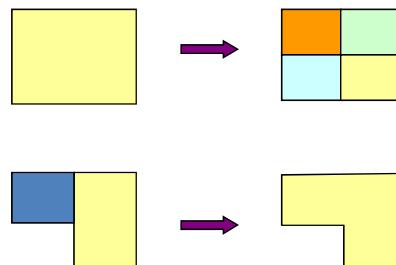
- Los umbrales no tienen en cuenta las características de la distribución espacial global.

- El método de **división y fusión de regiones (splitting and merging)**: divide (*split*) la imagen en un conjunto de regiones y después fusiona (*merge*) las que satisfacen algún criterio de similitud.
- Proceso:
 - Partición **inicial**: Todos los píxeles pertenecen a **una única región (R)**.
 - **Se define un criterio de similitud**: **distanza** de características de los píxeles (predicados) respecto a un modelo.

$$C = \sum_{p \in R_i} \|P(p) - M[P(R_i)]\|$$

p: píxel
 $P(\cdot)$: predicados calculados
 R_i : región
 $M[\cdot]$: modelo de la región (media, ...)

- **Fase split**: Si $C(R_i) = \text{falso}$, $C > T \rightarrow \text{Split}$, se divide cada región de forma geométrica. Se aplica a todas las regiones que se crean.
 La división geométrica genera mucha **sobresegmentación**.
- **Fase merge**: fusionar R_i y R_j .
 Si $C(R_i \cup R_j) = \text{verdadero}$, $C < T \rightarrow \text{merge}$.
 La fase de *merging* elimina la **sobresegmentación** producida en la fase de *splitting*.



□ Ejemplo de funcionamiento (I)

- Definición del criterio de similitud: $C_{R_i} = |I(p_j) - I(R_i)_{max}|; \forall p_j \in R_i (Th = 3)$

Split si $C_{R_i} \geq Th$

Quadimages

0	1	4	8
1	2	7	7
3	2	8	7
2	2	1	0

$C_R = 8$

0	1	4	8
1	2	7	7
3	2	8	7
2	2	1	0

$$C_{R_1} = 2$$

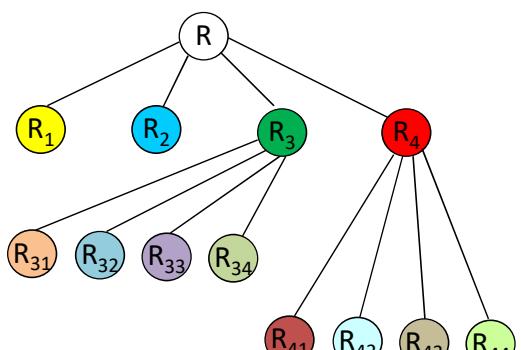
$$C_{R_2} = 1$$

0	1	4	8
1	2	7	7
3	2	8	7
2	2	1	0

$$C_{R_3} = 4$$

$$C_{R_4} = 8$$

Quadtree



Fin de la fase de splitting:

todas las regiones cumplen el criterio: Ya no es necesario realizar más divisiones.

□ Ejemplo de funcionamiento (II)

Merge si $C_{R_i} \cap C_{R_j} \leq Th$

0	1	4	8
1	2	7	7
3	2	8	7
2	2	1	0

0	1	4	8
1	2	7	7
3	2	8	7
2	2	1	0

0	1	4	8
1	2	7	7
3	2	8	7
2	2	1	0

0	1	4	8
1	2	7	7
3	2	8	7
2	2	1	0

$$C_{R_1} \cap C_{R_2} = 3$$

$$C_{R_1} \cap C_{R_{43}} = 2$$

$$C_{R_1} \cap C_{R_{44}} = 3$$

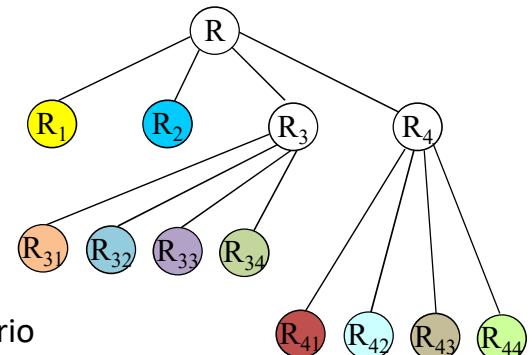
0	1	4	8
1	2	7	7
3	2	8	7
2	2	1	0

...

0	1	4	8
1	2	7	7
3	2	8	7
2	2	1	0

$$C_{R_{34}} \cap C_{R_{33}} = 1$$

$$C_{R_{34}} \cap C_{R_{41}} = 0$$



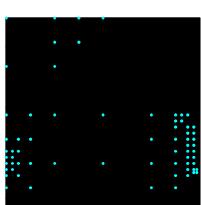
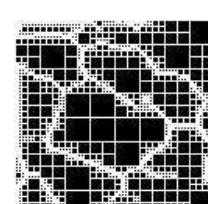
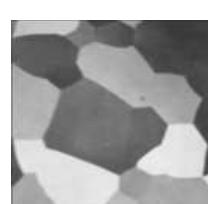
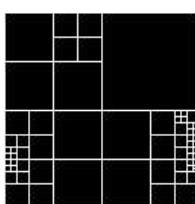
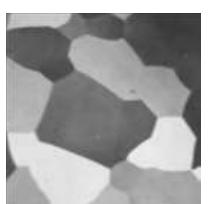
Seguimos hasta que ninguna unión cumpla el criterio para hacer el *merge*, eliminando la sobresegmentación.

□ Ventajas:

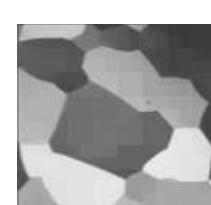
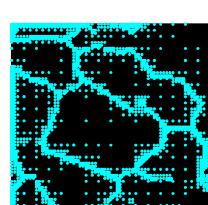
- Simple.
- Sin partición inicial fija.
- Visión global.

□ Inconvenientes:

- Dependencia del umbral.
- División puramente geométrica.
- Contornos poco naturales.



Umbral=0.6

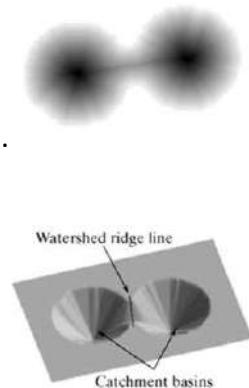


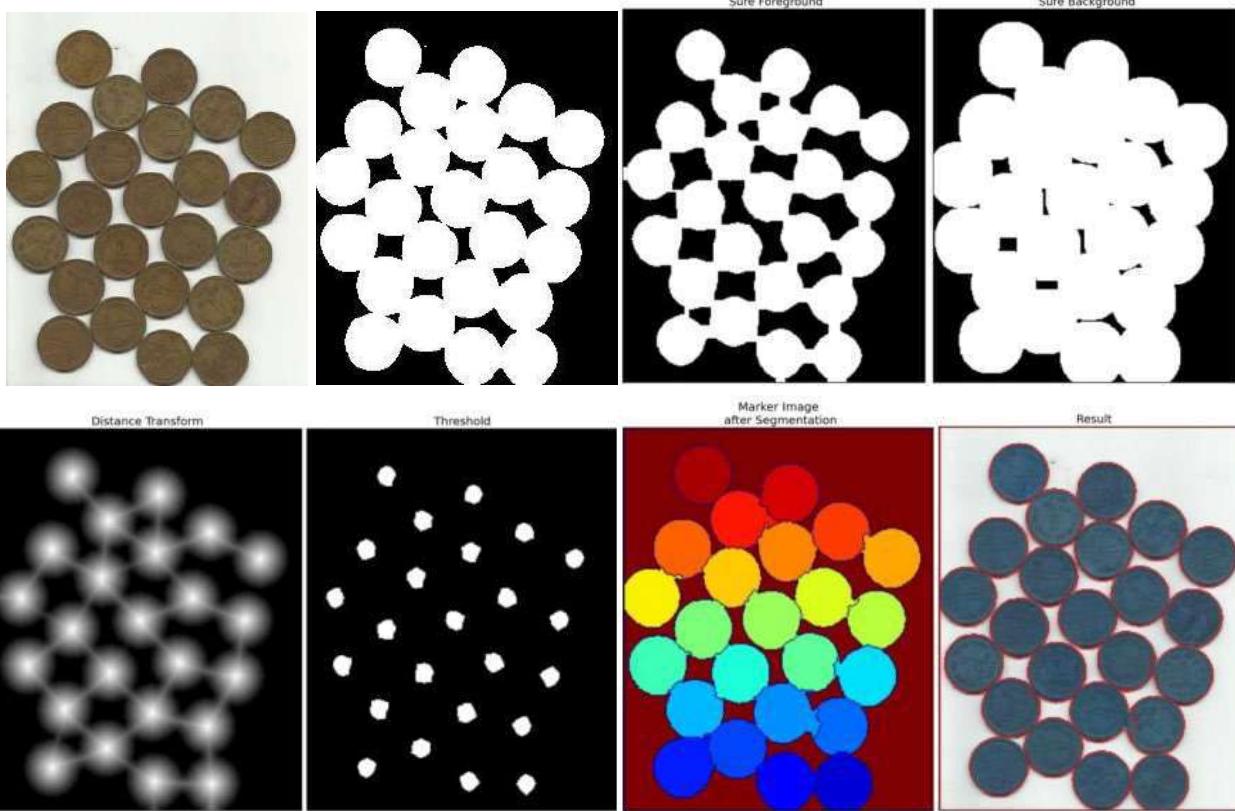
Umbral=0.1

- <http://www-2.cs.cmu.edu/afs/cs/project/cil/ftp/html/vision.html>
(Computer Vision Homepage CMU)
- <http://svr-www.eng.cam.ac.uk/milab.html> (Machine Intelligence Laboratory. Cambridge University)
- http://www.inria.fr/recherche/equipes/listes/theme_3.en.html
(INRIA Vision)
- <http://www-white.media.mit.edu/vismod/> (MIT Media Lab)
- <http://www.cvc.uab.es> (Centro de Visión Artificial. Barcelona)

Segmentación basada en regiones Transformada Watershed

- Transformada Watershed.** Es útil para separar **objetos cóncavos que se tocan** en una imagen (no se pueden separar por otros métodos).
- Aplica conceptos **topológicos** para segmentar. El algoritmo simplificado es:
 - Se **umbraliza** la imagen en niveles de gris, para obtener una imagen en blanco y negro.
 - A partir de ella se genera una imagen cuyo nivel de gris representa **la distancia de cada pixel de la imagen umbralizada al borde más cercano** del objeto. Interpretación topográfica: Niveles de gris altos (en la imagen de distancia) indican puntos muy alejados de los bordes. Se interpreta esa distancia como **profundidad**.
 - Se localizan los centros de los objetos (máximos locales de las distancias) umbralizando: valles (**catchment basins**, cuencas hidrográficas).
 - Se llena cada valle con agua de un color diferente.
 - Cuando el agua subiera obviamente diferentes colores se fusionarían.
 - Para evitarlo se construyen **barreras** (**watershed ridge**).
 - Se termina cuando todos los valles están bajo el agua, habiendo construido las barreras necesarias para que no fusione ninguno.
 - Las **barreras** creadas son el resultado de la segmentación.
- Problema:** sobresegmentación (a veces, se segmentan muchos más objetos de los necesarios, por ejemplo cuando los objetos tienen texturas que generan mínimos locales).
- Solución** a la sobresegmentación: preprocesar la imagen antes de aplicar el algoritmo con operadores morfológicos (erosión, dilatación, etc), marcar a "grosso modo" los objetos y el fondo, de cara a hacer la transformada con métodos automáticos (*fuzzy inference*, p.ej.).





□ **Ventajas:**

- Buen funcionamiento para segmentar múltiples objetos que se tocan aunque todos tengan similares niveles de gris (no podrían segmentarse con un crecimiento de regiones, ni por otros métodos).

□ **Inconvenientes:**

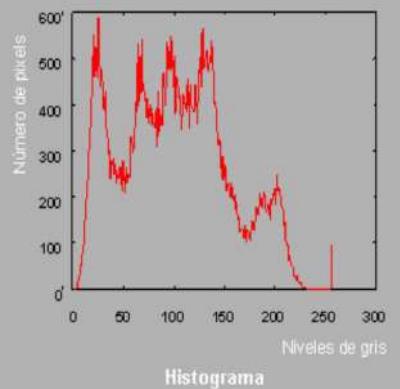
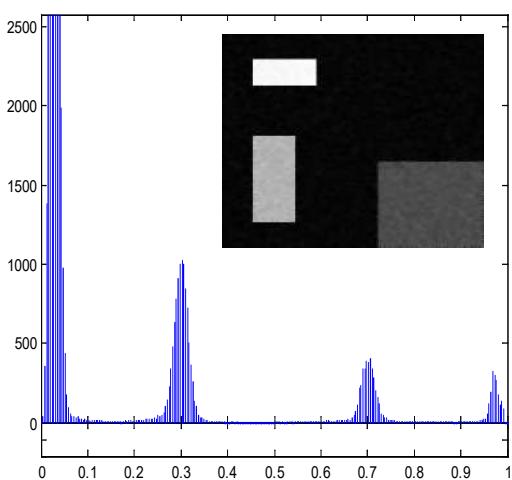
- Método complejo con gran contenido heurístico.
- Baja precisión en la posición de los contornos.



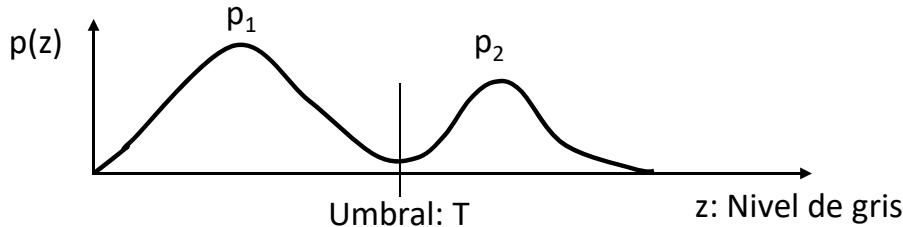
3. Segmentación basada en el histograma

Segmentación basada en histograma

- En ocasiones, los distintos elementos de interés de la imagen son identificables en el **histograma** → podemos utilizar esta característica en la **segmentación**.
- La complejidad del algoritmo necesario para dicha segmentación dependerá del grado de fusión de los histogramas de los distintos objetos.



- La segmentación basada en el **histograma** considera que cada **lóbulo** del histograma corresponde a la distribución en probabilidad de cada clase **objeto** en la imagen. Un ejemplo con 2 objetos podría ser:



- La segmentación se basa en un proceso de umbralización:

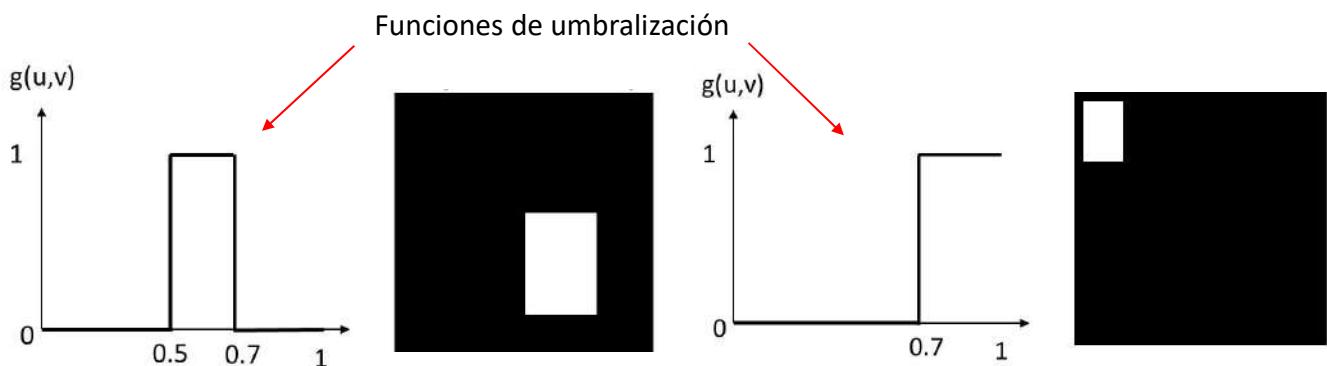
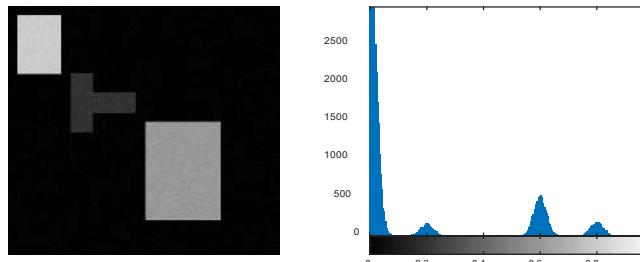
$$g(u,v) = \begin{cases} 255 & \text{si } f(u,v) > T \\ 0 & \text{si } f(u,v) \leq T \end{cases}$$

$f(u,v) \longrightarrow \boxed{\text{Umbralización}} \longrightarrow g(u,v)$

- **Problema: Cálculo del umbral $T = T [u, v, s(u,v), f(u,v)]$**
- $f(u,v)$ es la intensidad del píxel (u,v) y $s(u,v)$ es alguna propiedad local del píxel, por ejemplo, la intensidad media de un entorno de vecindad centrado en (u,v) .

- **Tipos de umbralización** (veremos ejemplos de todos en las siguientes transparencias):
 - Según el **número de umbrales** a aplicar en la imagen:
 - **Global**: mismos umbrales en toda la imagen → T depende sólo de $f(u,v)$.
 - **Local**: múltiples umbrales → T depende de $f(u,v)$ y $s(u,v)$.
 - **Multiespectral**: imágenes RGB → un umbral T para cada componente de color.
 - Según la **forma de cálculo**:
 - **Supervisados** (obtenidos manualmente):
 - **Método lineal**: Segmentación mediante distancia euclídea.
 - **Métodos no lineales**: Segmentación mediante el teorema de Bayes.
 - **No supervisados** (obtenidos de forma automática)
 - Búsqueda de **máximos y mínimos** sobre el histograma: Poco efectivo en imágenes reales (muchos máximos y mínimos locales).
 - Métodos de *clustering* (lineal): **Algoritmo K-means**.
 - Métodos estocásticos (no lineal): **Modelos de mezcla Gaussianos (GMM)**.

- A partir del histograma **global** de la imagen se obtienen los umbrales a aplicar a **todos** los píxeles de la imagen:



- En ocasiones, los valores de intensidad de los píxeles pueden **variар** en función de la iluminación de la escena: Es muy difícil/imposible segmentar **todas** las zonas de la imagen adecuadamente utilizando el mismo umbral global.
- Los resultados mejoran calculando un **umbral local, adaptativo**, diferente para cada entorno, definido para una vecindad del píxel a umbralizar, en función de algún estadístico del entorno como: media, mediana, etc.

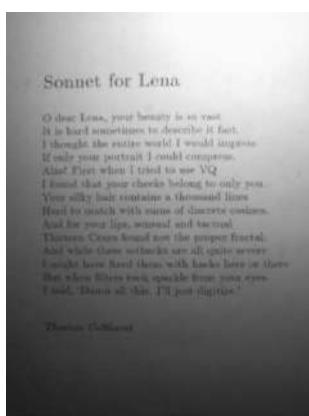


Imagen original



Imagen umbralizada
con umbral global

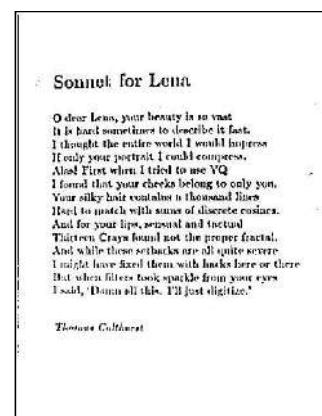
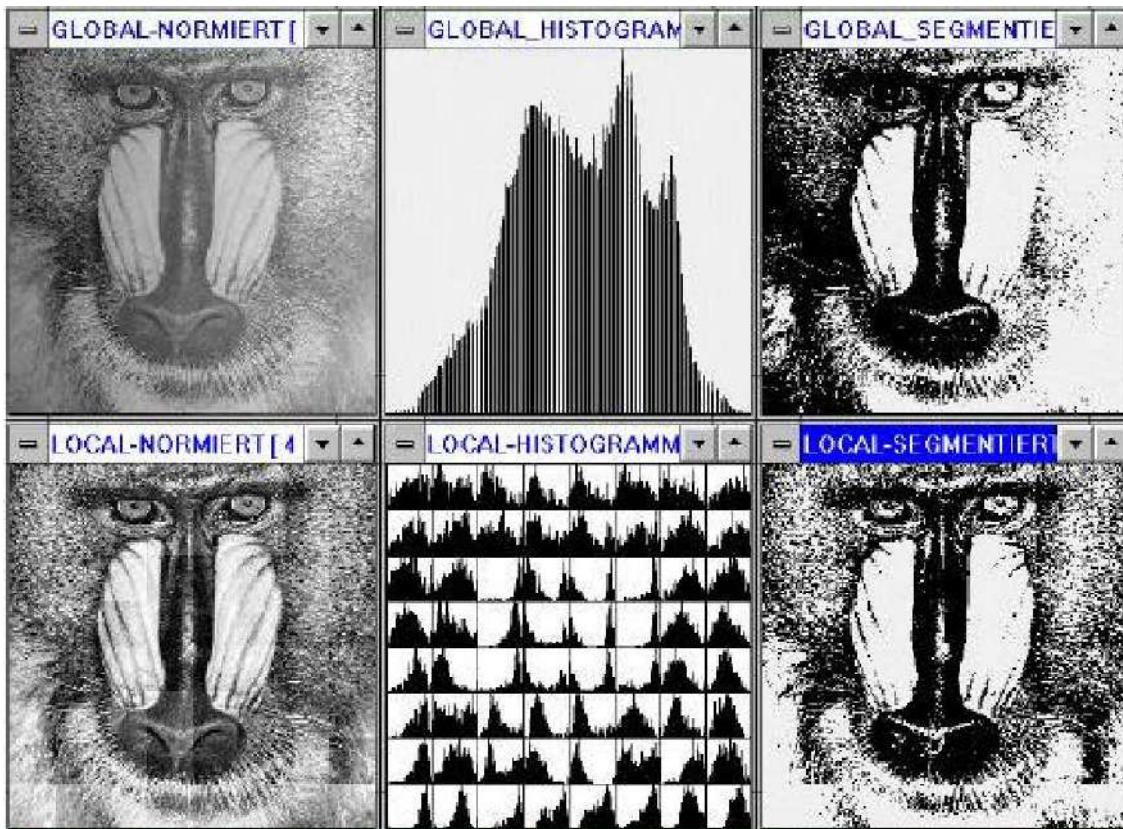
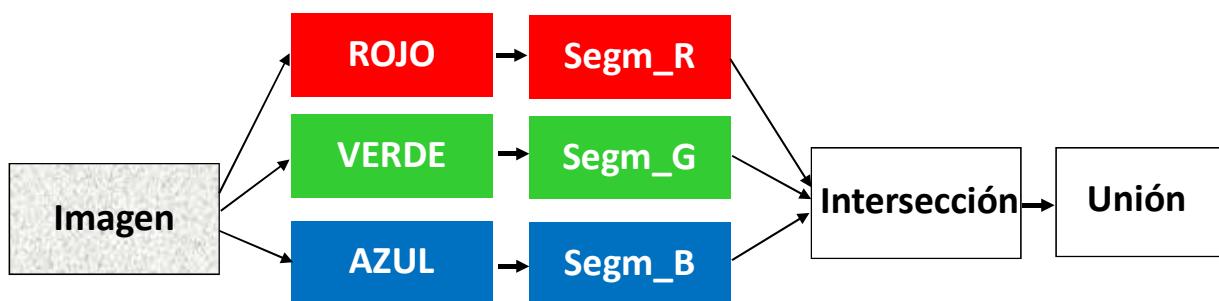
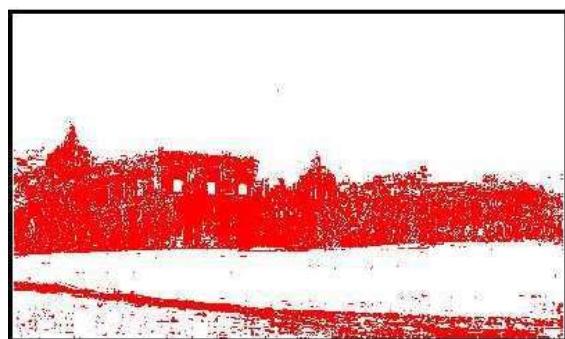
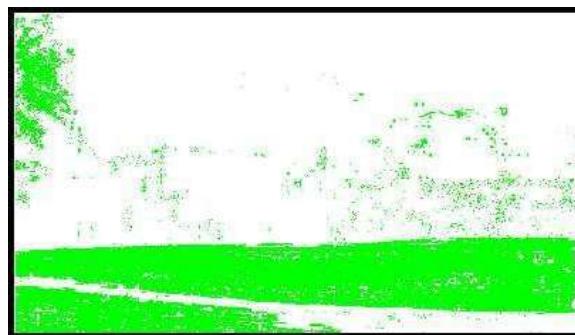


Imagen umbralizada
con umbral local

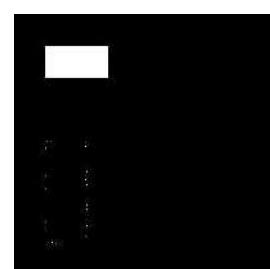
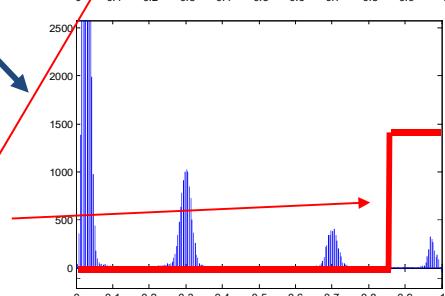
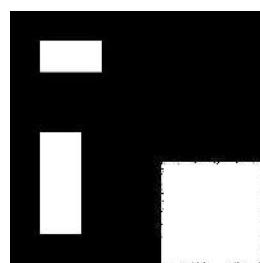
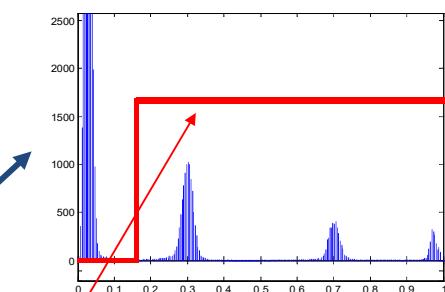
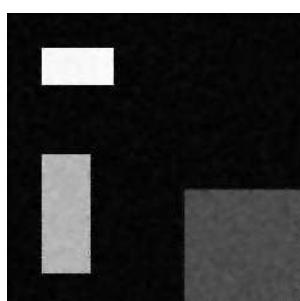


- Cuando la imagen es en **color**, se puede realizar una **umbralización multiespectral** que se consigue mediante la unión (OR) o intersección (AND) de las imágenes umbralizadas para **cada componente R, G, B**.



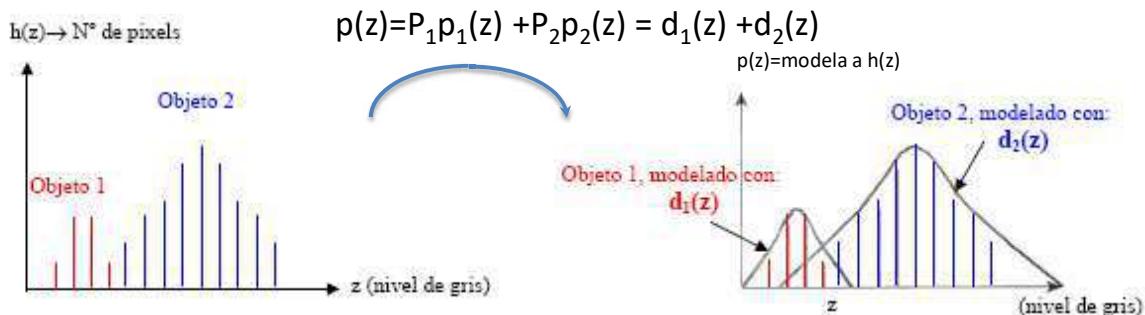


- **Umbralización supervisada:** hay un **entrenamiento** previo en que se proporciona al sistema información sobre las clases y sus características.
- **Umbralización supervisada lineal:** Se obtienen los **umbrales** a partir de los **centroides** de las clases buscando (manualmente o calculando) los puntos medios entre ellos: **segmentación por distancia euclídea**.



Funciones de umbralización

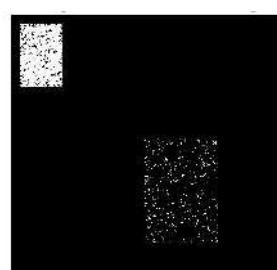
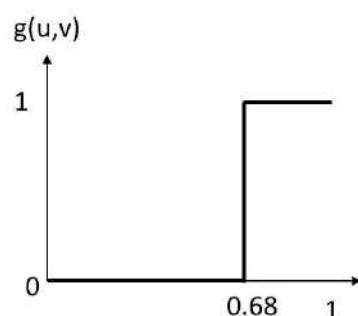
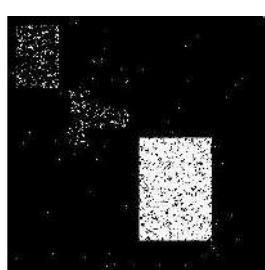
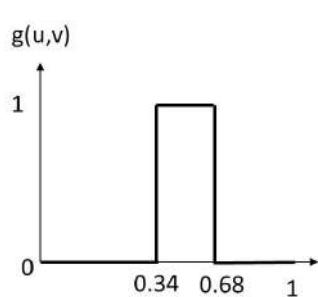
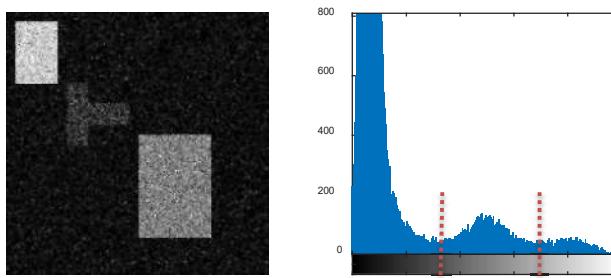
- **Umbralización supervisada no lineal:** hay veces que los objetos tienen histogramas diferenciados en distintos lóbulos, pero el umbral óptimo **no** es el punto medio.
- Se aproxima el histograma mediante una **suma ponderada de funciones de densidad de probabilidad** (fdp o pdf en inglés) correspondientes a los niveles de gris de los objetos presentes en la imagen. Para el caso de un histograma bimodal, dos objetos (1 y 2) con lóbulos bien diferenciados:



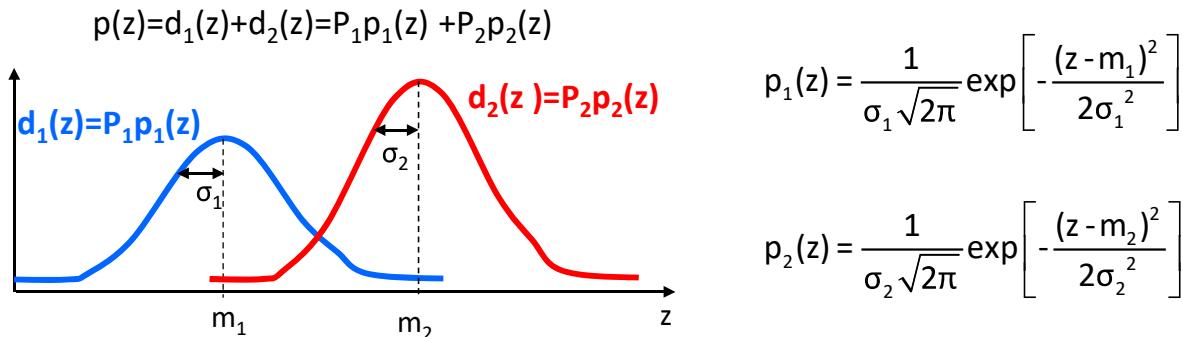
Donde:

z = variable aleatoria que representa intensidad luminosa (nivel de gris) en la imagen,
 $p_1(z)$ y $p_2(z)$ = fdp asociadas a los dos objetos del histograma,
 P_1 y P_2 = probabilidades a priori de que un píxel pertenezca al objeto 1 y al 2 ($P_1+P_2=1$).
 $d_1(z) = P_1p_1(z)$ y $d_2(z) = P_2p_2(z)$: modelos del histograma referentes al objeto 1 y 2.

- Cuando los lóbulos de los histogramas de los objetos están **mezclados** (es lo normal), el proceso de segmentación **no es perfecto**: se pueden segmentar correctamente la mayoría de los píxeles, pero algunos se clasifican **erróneamente**.



- Es frecuente hacer la **aproximación por funciones Gaussianas**:



donde:

m_1, σ_1 y m_2, σ_2 = media y desviación típica de las fdps asociadas a los objetos 1 y 2.

- Los **pesos**, P_1 y P_2 , que ponderan las fdp serán proporcionales al área que ocupa la parte del histograma referente al objeto 1 y 2:

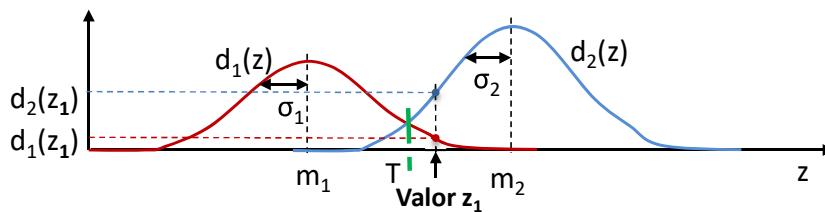
- más probabilidad de objeto 2 que de objeto 1: $P_2 > P_1$
- clases equiprobables: $P_1 = P_2 \dots$

$$\left. \begin{array}{l} P_1 + P_2 = 1 \end{array} \right\}$$

- Para cada píxel, se toma su nivel de gris, z_1 , se calcula $d_i(z_1)$ para cada la fdp de cada objeto y se asigna el pixel a la clase que proporcione un valor mayor de $d_i(z_1)$, es decir, cada píxel se clasifica en la clase en la que la **pertenencia al modelo** de su nivel de gris es **máxima**:

Si $d_1(z_1) > d_2(z_1)$ el píxel pertenece al objeto 1

Si $d_2(z_1) > d_1(z_1)$ el píxel pertenece al objeto 2



- El **umbral óptimo** se selecciona para aquel nivel $z = T$ que cumple:

$$d_1(z) = d_2(z) \rightarrow P_1 * p_1(z) = P_2 * p_2(z)$$

- Si $\sigma_1 = \sigma_2 = \sigma$ (histograma bimodal modelado con funciones gaussianas):

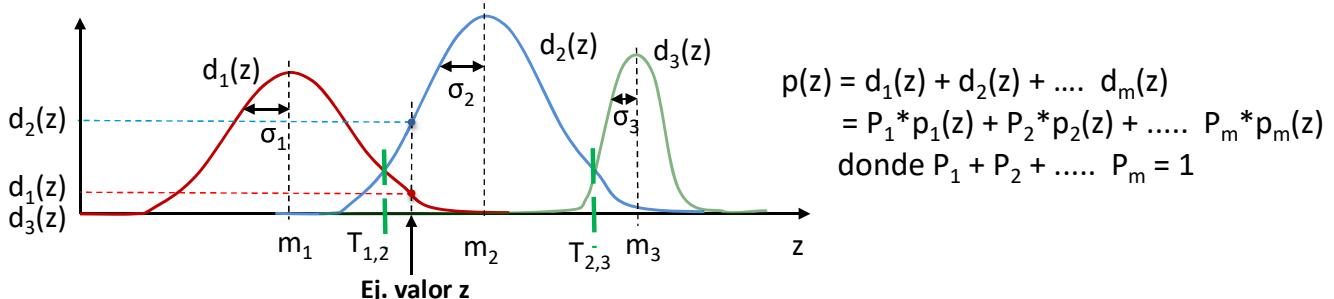
$$T = \frac{m_1 + m_2}{2} + \frac{\sigma^2}{m_1 - m_2} \ln \frac{P_2}{P_1}$$

- Calculado el umbral, no es necesario aplicar la fórmula de cada gausiana, el proceso de clasificación se reduce a la comparación del nivel de gris con dicho umbral:

Si $z_1 < T$ el píxel pertenece al objeto 1

Si $z_1 > T$ el píxel pertenece al objeto 2

- Para **histogramas multimodales** (con m objetos en la imagen):



- Cada píxel se clasifica como perteneciente a la clase para la cual la pertenencia al modelo de su nivel de gris (z) sea **máxima**. El píxel de nivel de gris z pertenece al objeto o clase *k*-ésima si: $d_k(z) > d_j(z)$, siendo $j=1,2\dots m$ y distinto de *k*.
- Los **umbrales** entre clase y clase se seleccionan de forma que:

$$P_k * p_k(T_{kj}) = P_j * p_j(T_{kj})$$

siendo T_{kj} el umbral de separación entre la clase *k* y *j* contiguas.

- Una vez elegidos los umbrales, no será necesario aplicar las fórmulas de las gaussianas: compararemos el nivel de gris con los distintos umbrales. **El píxel de nivel de gris z pertenece al objeto o clase *k*-ésima si $T_{k-1} < z < T_k$**

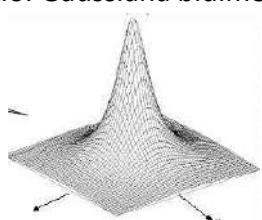
- Este procedimiento se puede generalizar para un **histograma n-dimensional**, (que representa el nº de puntos de la imagen con un vector n-dimensional de características de valor igual a \bar{z}), **de m objetos**, que se modelaría mediante:

$$p(\bar{z}) = d_1(\bar{z}) + d_2(\bar{z}) + \dots + d_m(\bar{z}) = P_1 * p_1(\bar{z}) + P_2 * p_2(\bar{z}) + \dots + P_m * p_m(\bar{z})$$

siendo: $P_1 + P_2 + \dots + P_m = 1$

$p_1(\bar{z}), p_2(\bar{z}) \dots p_m(\bar{z})$ fdp n-dimensionales asociadas con los objetos 1, 2.... m:

Ejemplo: Gaussiana bidimensional



$$p_1(\bar{z}) = \frac{1}{(2\pi)^{\frac{n}{2}} |C_1|^{\frac{1}{2}}} \exp \left[-\frac{1}{2} (\bar{z} - \bar{\mu}_1)^T C_1^{-1} (\bar{z} - \bar{\mu}_1) \right]$$

donde $\bar{\mu}_1, C_1$ representan los vectores medias y matrices de covarianza de la fdp asociada con la parte del histograma n-dimensional del objeto 1.

- El píxel de vector de características \bar{z} pertenecería a la clase *k*-ésima si: $d_k(\bar{z}) > d_j(\bar{z})$ siendo $j = 1, 2, \dots, m$ y distinto de *k*.

- Si se desea **segmentar un único objeto** con un color uniforme en la imagen se puede modelar la fdp de este color mediante una función Gaussiana definida en el espacio de color (espacio RG):
 - Cálculo de los parámetros del modelo a priori
 - Segmentación mediante umbral



Imagen original



Objeto a segmentar

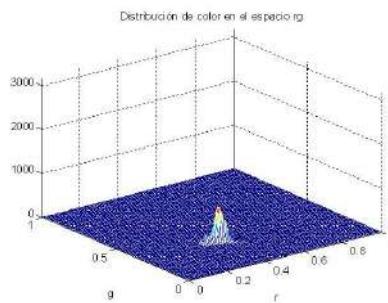


Imagen segmentada

$$f(x_j / \text{skin}) = \frac{1}{2\pi |C_s|^{0.5}} e^{-0.5(x_j - m_s)^T C_s^{-1} (x_j - m_s)}$$

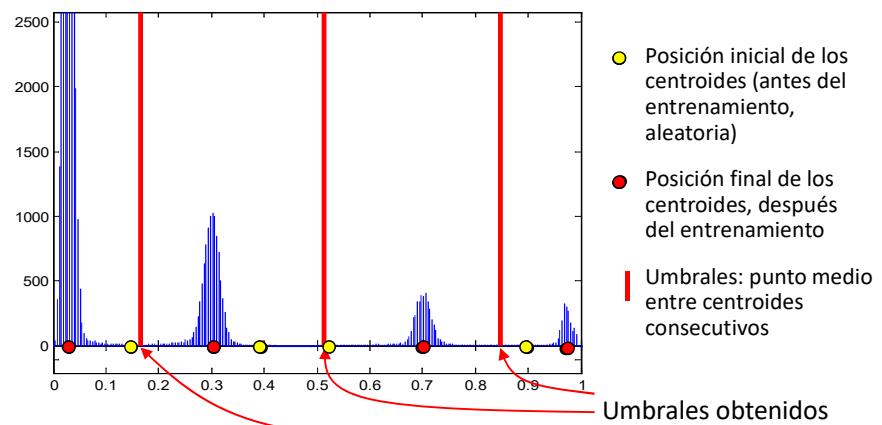
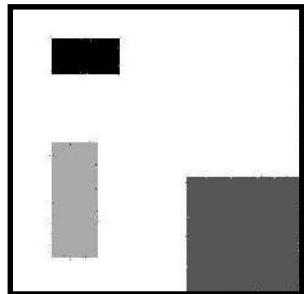
$x_j \in \text{skin}$ si $f(x / \text{skin}) > \text{Th}$

Segmentación

m_s
 C_s

Parámetros del
modelo

- **Algoritmos de clustering:** clasificación automática, no supervisada, de los píxeles: el sistema aprende automáticamente, a partir de un conjunto de entrenamiento en que no le damos previamente clasificados los niveles.
- **Umbralización no supervisada lineal:**
 - Número de clases conocido a priori,
 - Entrenamiento no supervisado de centroides (puntos promedio de cada clase),
 - Clasificación mediante distancia euclídea (lineal): umbrales en el punto medio entre cada par de centroides.



□ Algoritmo *k-means*:

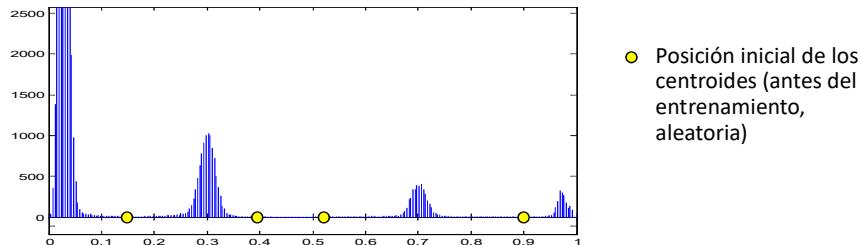
- Su objetivo es clasificar los píxeles $\{X\}$ de la imagen en m clases $\{\alpha_1, \alpha_2, \alpha_m\}$
- Es un algoritmo de clasificación **no** supervisada lineal, que asume que **se conoce el número de clases** a clasificar (m), pero nada más: en la etapa de entrenamiento no se conoce a qué clase pertenece cada nivel de gris.

□ Algoritmo de entrenamiento recursivo para encontrar el centroide de cada clase (I):

- **Paso 1:** Se toman al **azar** entre los elementos a agrupar m vectores, **centrodes de las m clases**.

$$\alpha_1:z_1(0); \alpha_2:z_2(0); \dots; \alpha_m:z_m(0)$$

- Las componentes de los vectores son características de los píxeles: nivel de gris, color, etc.



□ Algoritmo de entrenamiento recursivo para encontrar el centroide de cada clase (II):

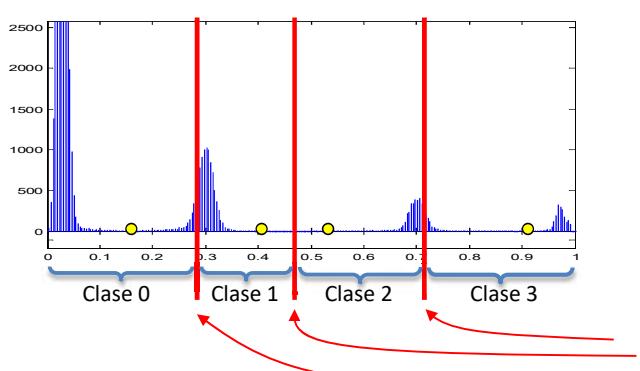
- **Paso 2:** En la iteración t **se distribuyen todas las muestras entre las m clases**, $\alpha_i(t)$

For $i = 1$ to N

$$x_i \in \alpha_j(t) \quad \text{si} \quad \|x_i - z_j(t)\| < \|x_i - z_k(t)\|$$

$$\forall k = 1, 2, \dots, m / k \neq j$$

End{For}



● Posición inicial de los centroides (antes del entrenamiento, aleatoria)

Umbrales: punto medio entre centroides consecutivos

Umbrales correspondientes con esos centroides

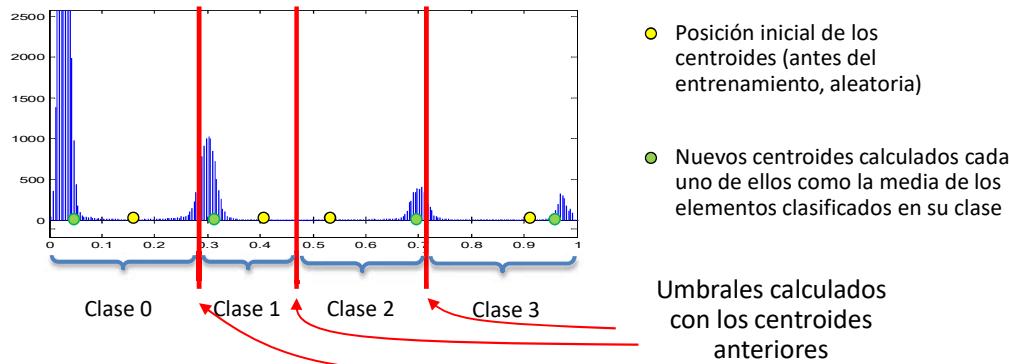
□ Algoritmo de entrenamiento recursivo para encontrar el centroide de cada clase (III):

- Paso 3: Se actualizan los centroides de las clases con objeto de minimizar la siguiente función de coste:

$$J_k = \sum_{x \in \alpha_k(t)} \|X - z_k(t)\|^2 \quad k = 1, 2, \dots, m \quad (m \text{ es el n}^{\circ} \text{ de clases})$$

Con esto conseguimos maximizar la variación *intercluster* y minimizar la variación *intraccluster*. Esta minimización se logra utilizando la media aritmética de $\alpha_k(t)$.

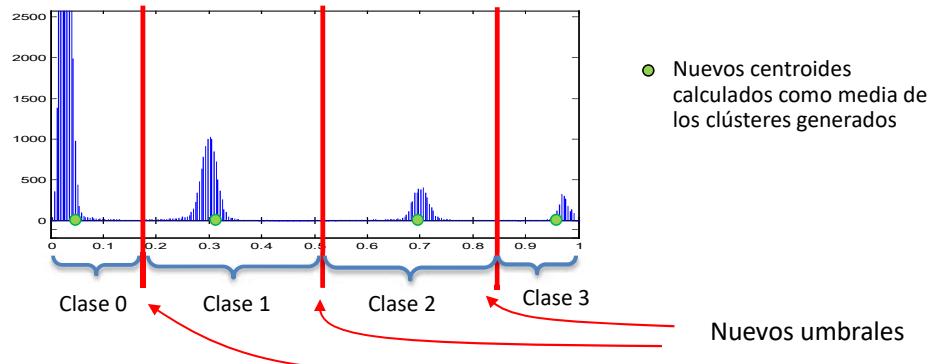
$$z_k(t+1) = \frac{1}{N_k(t)} \sum_{i=1}^{N_k(t)} x_i \quad ; k = 1, 2, \dots, m \quad ; N_k(t) : \text{elementos de } \alpha_k(t)$$



□ Algoritmo de entrenamiento recursivo para encontrar el centroide de cada clase (IV):

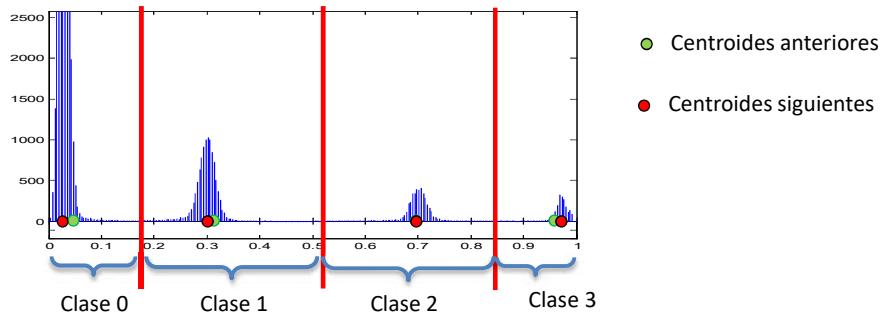
- Paso 4: Se vuelve al paso 2 y se repiten todos los pasos hasta que los vectores prototipo (centroides) no cambien su posición cuando se recalculen, o hasta que el error esté por debajo de un umbral:

$$\sum_{j=1}^m \|z_j(t+1) - z_j(t)\| < \epsilon$$

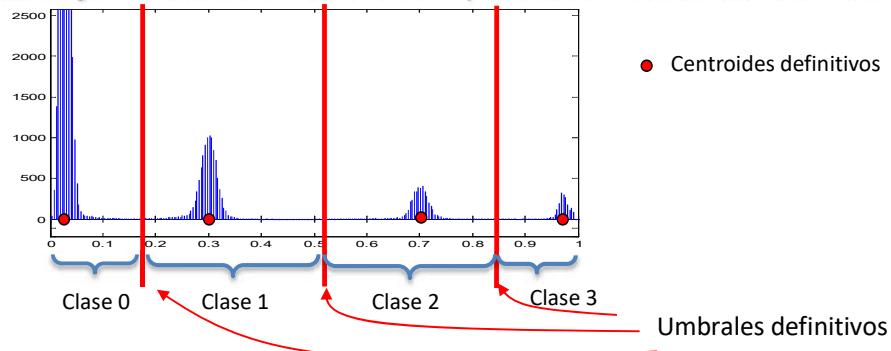


□ Algoritmo de entrenamiento recursivo para encontrar el centroide de cada clase (V):

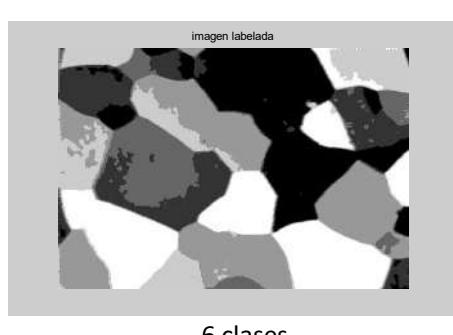
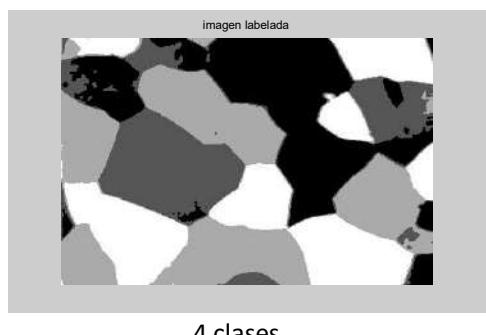
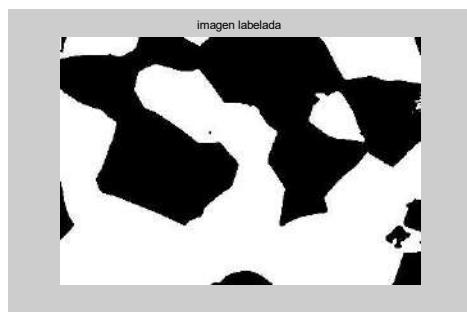
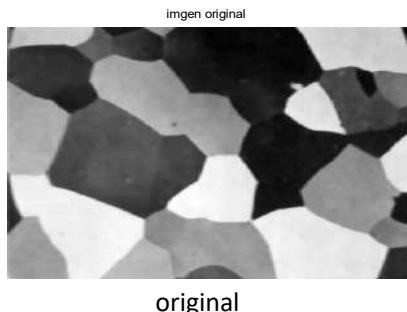
- Siguiente iteración:



- Valores definitivos: en la siguiente iteración no se modifica la posición de los centroides ni de los umbrales.



□ Ejemplos:



Ventajas:

- Algoritmo sencillo y muy eficiente.

 Inconvenientes:

- Se necesita saber el **número de clases** a priori.
- El método de cálculo de clases no es óptimo ya que considera que todas tienen **la misma dispersión estadística para cada dimensión (d)** de X , y esto no es cierto en todos los casos.
- Aprendizaje local y dependiente de la inicialización**, en cada iteración únicamente se mueve el centroide de la clase ganadora.

 Segmentación basada en histograma **Ventajas:**

- Aplicación **simple y rápida**
- Puede obtenerse de forma **no supervisada**
- Buenos resultados con imágenes **simples** e iluminación controlada
- Aplicable para aplicaciones de **seguimiento** de objetos en secuencias

 Inconvenientes:

- Usa un umbral **único** para toda la imagen: malos resultados con iluminación no uniforme
- No controla la conectividad espacial**
 - Agujeros** en las clases
 - Píxeles **aislados** de una clase dentro de otras

- [Bergasa et al.,00] L.M. Bergasa, M. Mazo, A. Gardel, M. A. Sotelo, L. Boquete, "Unsupervised and adaptative Gaussian skin-color model", Image and Vision Computing (Editorial Elsever), Number 18, pages 987 to 1003, September of 2000.
- [Campbell et al. 96] N.W. Campbell, B.T. Thomas and T. Troscianko. "Segmentation of Natural Images using Self_Organising Feature Maps". In British Machine Vision Conference, pages 223_232. British Machine Vision Association, September 1996.
- [Gath et al.,89] I. Gath and B. Geva. "Unsupervised Optimal Fuzzy Clustering". IEEE Trans. Pattern Analysis and Machine Intelligence", Vol.11, Nº7, pp.773-781, July 1989.
- [Kohonen, 97] T. Kohonen, "Self Organizing Maps", Springer, Berlin, 1997.
- [Lagan et al.,98] D.A. Lagan, J.W. Modestino and J.Zhang. "Cluster Validation for Unsupervised Stochastic Model-Bases Image Segmentation". IEEE Transactions on Image Processing, Vol. 7, Nº2, pp. 180-195, February, 1998
- [Littmann&Ritter, 97] E. Littmann and H. Ritter, "Adaptive Color Segmentation-A comparation of Neural and Statistical Methods", IEEE Transactions on Neural Networks, VOL 8, NO 1, January 1997.
- [Oliver et al.,00] N. Oliver, A. Pentland, F. Berard, "LAFTER: a real-time face and lips tracker with facial expression recognition". Pattern Recognition 33, pp:1369-1382, 2000.
- [Rissanen, 78] J. Rissanen. "Modeling by Shortest Data Description", Automatica, Vol.14, pp. 465-471, 1978.
- [Roberts et al., 98] S.J. Roberts, Dirk Husmeier, lead Rezek and William Penny. "Bayesian Approaches to Gaussian Mixture Modeling". IEEE Transactions on Image Processing, Vol. 20, Nº11, pp.1133-1142, November, 1998
- [Stiefelhagen et al., 97a] Stiefelhagen, J. Yang and A. Waibel "A model based gaze tracking system". IEEE International Joint Symposia on Intelligence and Systems-Image, Speech&Natural Language Systems. 1997
- [Yamazaki, 98] T. Yamazaki, "Introduction of EM algorithm into color image segmentation", Proc. ICIPS'98, pp:368-371, August, 1998.
- [Yang&Waibel, 97], J. Yang, A. Waibel, "Skin-Color Modelong and Adaptation", Technical Report CMU-CS-97-146. School of Computer Science. Carnegie Mellon University, 1997.
- [Zhu et al., 00] X. Zhu, J. Yang, A. Waibel, "[Segmenting Hands of Arbitrary Color](#)", Fourth IEEE International Conference on Automatic Face and Gesture Recognition (F&G),pp: 446-453, 2000