

"Diseño e Implementación de una Herramienta Didáctica de Software para el Procesamiento Digital Básico de Imágenes para los Estudiantes de la Facultad de Ingeniería en Electricidad y Computación"

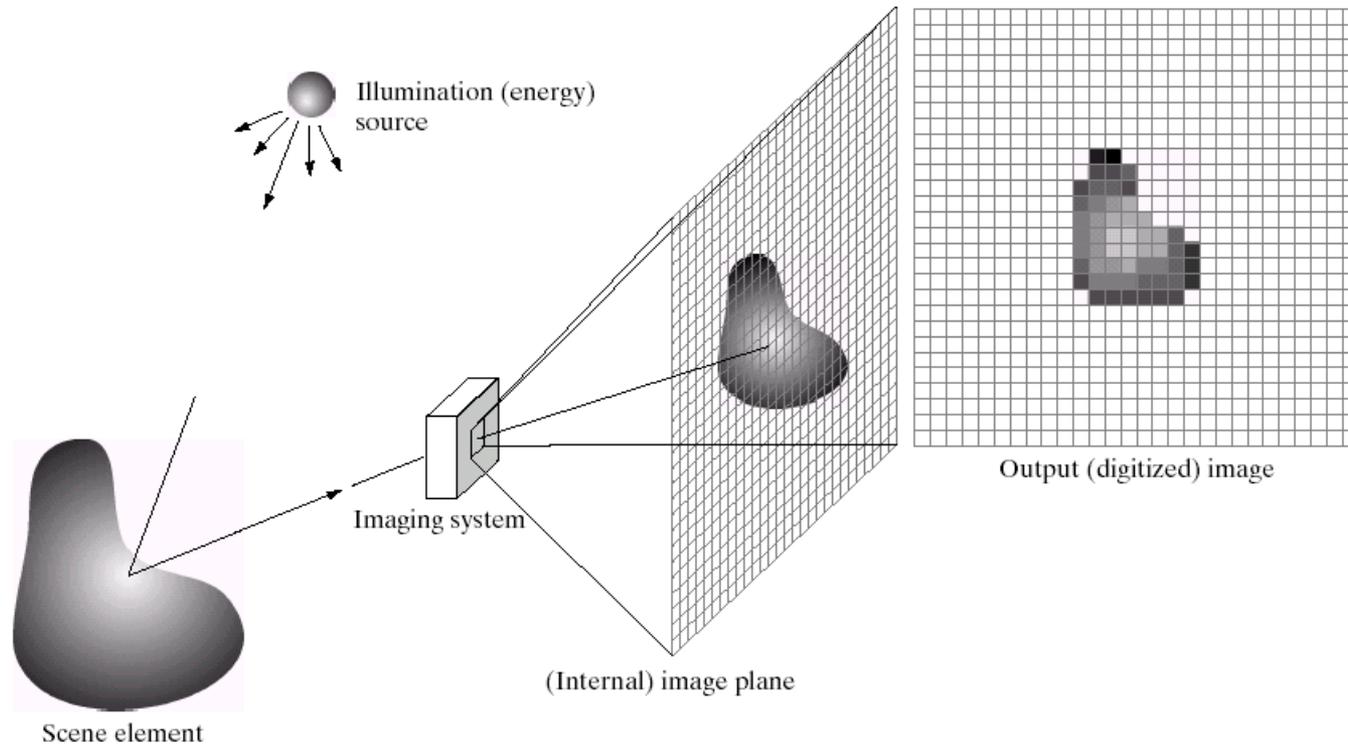
Por: Alex F. Guerrero E.

Fecha: 04/02/2010

Tipos de Imágenes:

- Una imagen natural capturada con una cámara, un telescopio, un microscopio o cualquier otro tipo de instrumento óptico presenta una variación de sombras y tonos continua. Imágenes de este tipo se llaman ***imágenes analógicas***.
- Para que una imagen analógica, en blanco y negro, en escala de grises o a color, pueda ser "manipulada" usando un computador, primero debe convertirse a un formato adecuado. Este formato es la ***imagen digital*** correspondiente.
- La transformación de una imagen analógica a otra discreta se llama ***digitalización*** y es el primer paso en cualquier aplicación de procesamiento de imágenes digitales.

Captura de Imagen



a b c d e

FIGURE 2.15 An example of the digital image acquisition process. (a) Energy (“illumination”) source. (b) An element of a scene. (c) Imaging system. (d) Projection of the scene onto the image plane. (e) Digitized image.

Clasificación de imágenes digitales:

- Clasificación de imágenes digitales:
 - Por dimensión: Imágenes 2D y 3D
 - Por paleta de colores: imágenes binarias, en escala de grises y a color.

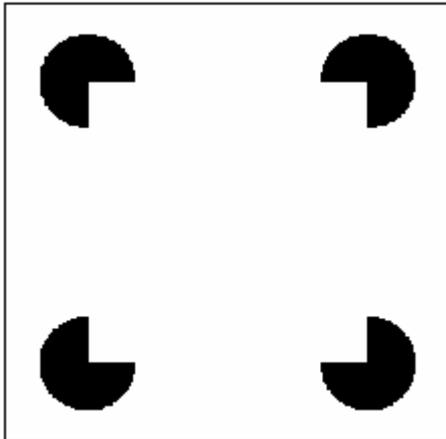


Imagen binaria



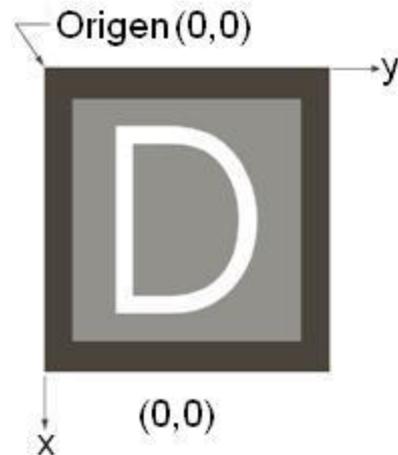
Imagen en escala de grises



Imagen a color

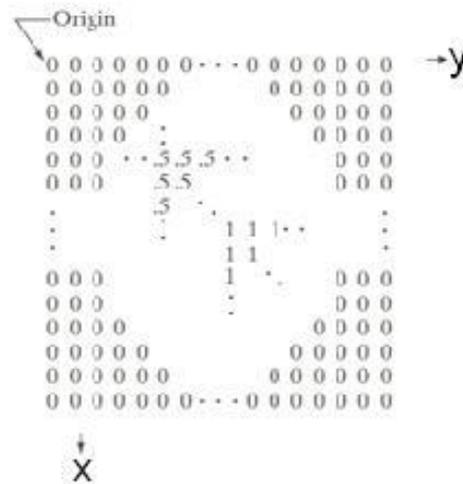
Imágenes en 2D

- Nos centraremos en imágenes digitales cuadradas o rectangulares, cuyos píxeles (x,y) representan regiones cuadradas.
- La coordenada x especifica la fila donde está localizado el píxel; la coordenada y representa la columna.
- Por convención, el píxel $(0,0)$ está localizado en la esquina superior izquierda de la imagen.



Imágenes en 2D

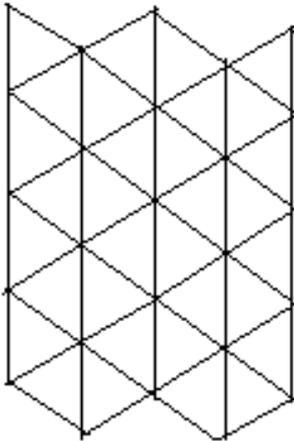
- Una imagen digital de $M \times N$ píxeles en escala de grises (con L niveles de gris) es una función
 - $f: [0, M-1] \times [0, N-1] \rightarrow [0, L-1]$,tal que a cada punto (píxel) (x, y) , le asigna un valor (nivel de gris).



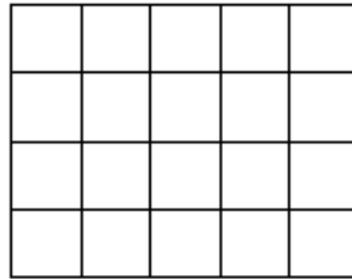
- La cuantificación consiste en una paleta de 256 niveles de gris (donde 0 indica el color negro y 255 el color blanco):

Muestreo

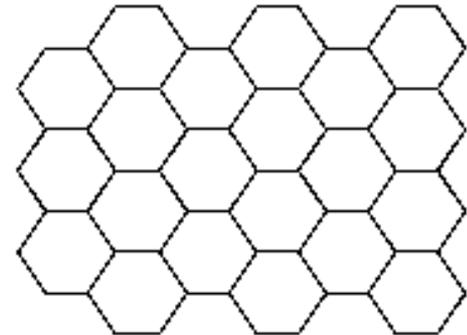
- Consiste en una subdivisión de la imagen analógica en porciones. Nos centraremos en imágenes 2D. Existen particiones que envuelven polígonos regulares: triángulos, cuadrados y hexágonos.



Mallado triangular



Mallado cuadrangular



Mallado hexagonal

Cuantificación

- La salida de estos sensores es un valor (**amplitud**) dentro de una escala (**color**). La salida pueden ser, o bien un único valor (**escala de grises**) o bien un vector con tres valores por polígono (**RGB**) que se corresponden con la intensidad de color rojo (R), verde (G) y azul (B). La escala de colores también tiene un rango discreto (por ejemplo, de 8-bits = 256 valores).

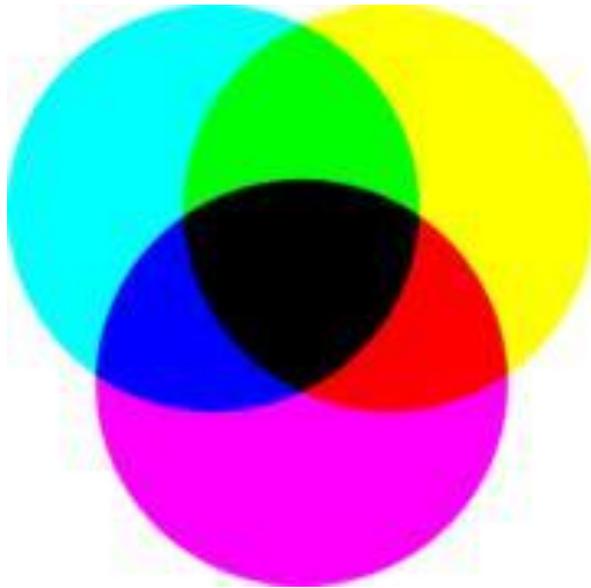
Las imágenes en escala de grises con sólo 2 colores: blanco y negro (0 y 1, respectivamente), se llaman *imágenes binarias*.

A este proceso de discretización del color se le llama **cuantificación**.

Un polígono de color constante se llamará *píxel*.

Colores RGB y CMY

- El **cian** es el opuesto al [rojo](#), lo que significa que actúa como un filtro que absorbe dicho color ($-R +G +B$). **Magenta** es el opuesto al [verde](#) ($+R -G +B$) y **amarillo** el opuesto al [azul](#) ($+R +G -B$).



Imágenes a color: Modelo RGB

- Las imágenes digitales a color están gobernadas por los mismos conceptos de muestreo, cuantificación y resolución que las imágenes en escala de grises.
- Sin embargo, en lugar de un único valor de intensidad que expresa el nivel de gris, los píxeles de las imágenes a color están cuantificados usando tres componentes independientes uno por cada color primario (**RGB = rojo, verde y azul**).
- Combinando distintas intensidades de estos tres colores, podemos obtener todos los colores visibles.

Modelo CMY

- En algunos casos, son más apropiados modelos diferentes del RGB para algoritmos y aplicaciones específicas. De cualquier manera, cualquier otro modelo sólo requiere una conversión matemática simple para obtener el modelo RGB.
- **Para imprimir** una imagen digital, es necesario convertir la imagen RGB al modelo CMY (**cian-magenta-amarillo**).
- La conversión es:

$$\begin{pmatrix} R \\ G \\ B \end{pmatrix} = \begin{pmatrix} L \\ L \\ L \end{pmatrix} - \begin{pmatrix} C \\ M \\ Y \end{pmatrix}$$

siendo $L+1$ es la cantidad de niveles de color de la imagen.

Ej. Conversión de colores

- Imagen comparativa en la que se observan las diferencias en el color entre el modelo RGB (izquierda) y el modelo CMYK (derecha).



Modelo YIQ

- El modelo YIQ se usa en las **televisiones** comerciales.
- Básicamente, YIQ es una recodificación de RGB para mantener la compatibilidad con las televisiones en blanco y negro. De hecho, la componente **Y (*luminancia*)** provee toda la información requerida para una televisión en blanco y negro. La conversión de RGB a YIQ es:

$$\begin{pmatrix} Y \\ I \\ Q \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.275 & -0.321 \\ 0.212 & -0.523 & 0.311 \end{pmatrix} * \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

- Si sólo tenemos en cuenta la componente Y de la imagen, lo que obtenemos es una imagen en escala de grises. Así pues, la forma de **obtener una imagen en escala de grises a partir de una en RGB** es aplicando al valor RGB de cada píxel, la fórmula

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

Modelo HSI

- Otro modelo muy utilizado es el HSI que representa el color de una manera intuitiva (es decir, de la forma en los humanos percibimos el color). La componente I se corresponde con la **intensidad**, H con el **color** y S con la **saturación**. Este modelo es muy utilizado en algoritmos de procesamiento de imágenes basados en propiedades del **sistema de visión humano**.
- La conversión de RGB a HSI es más complicada. Pero la componente I es fácil de calcular:

$$I = 1/3 * (R+G+B)$$

Filtros y Detectores de Borde

Filtro definición

- Se le llama filtrado al proceso mediante el cual se modifica una señal determinada de tal manera que las amplitudes relativas de las componentes en frecuencia cambian o incluso son eliminadas.



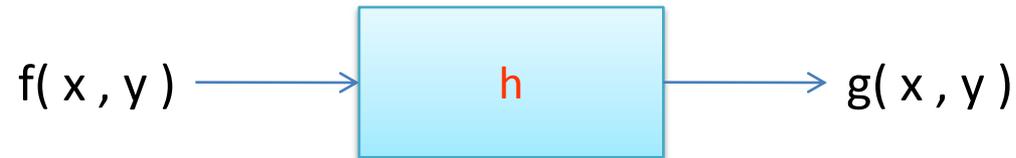
- También sirven para restaurar una señal, cuando haya una señal que haya sido deformada de alguna forma.
- La función en Matlab que permite generar un filtro para aplicarlo sobre una imagen en 2-D es *fspecial*, y contiene filtros predefinidos en 2-D.

Filtros en el dominio del espacio:

- Filtros
 - Filtros espaciales
 - Filtros lineales
 - Filtros pasa bajos
 - Filtro pasa altos
 - Filtro pasa bandas
 - Filtros no lineales
 - Filtro max
 - Filtro min
 - Filtro mediana

Filtro espacial

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



- También se le denomina procesamiento basado en la vecindad u operación de vecindad.
- Filtrar una imagen consiste en aplicar una transformación de forma que se acentúen o disminuyan ciertos aspectos.

Tipos de Filtros

Filtro espacial - convolución

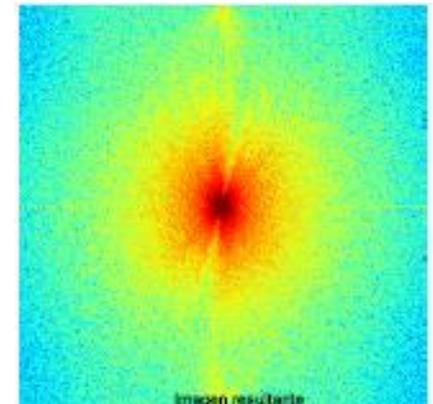
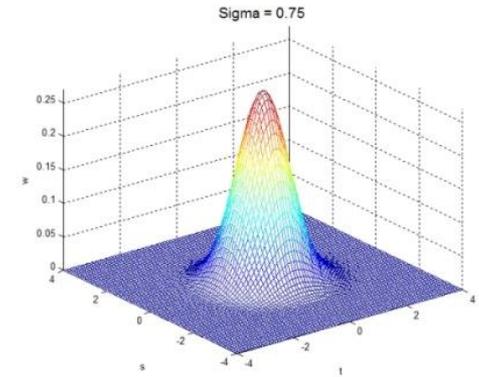
- La alteración del píxel se realiza dependiendo de los valores de los píxeles del entorno sin realizar ninguna modificación previa de sus valores

$$g(x, y) = h(x, y) * f(x, y)$$

Filtrado frecuencial - multiplicación + transformadas de Fourier

- Requiere de la aplicación de la transformada de Fourier.

$$G(u, v) = H(u, v) F(u, v) \quad g(x, y) = T[f(x, y)]$$



Generación de Filtros en Matlab

`h = fspecial(type)`

`h = fspecial(type, parameters)`

Crea un filtro bidimensional **h** del tipo especificado.

Devuelve como un kernel de correlación, que es la forma adecuada para usar con *imfilter*.

type	description
'average'	Averaging filter
'disk'	Circular averaging filter (pillbox)
'gaussian'	Gaussian lowpass filter
'laplacian'	Approximates the two-dimensional Laplacian operator
'log'	Laplacian of Gaussian filter
'motion'	Approximates the linear motion of a camera
'prewitt'	Prewitt horizontal edge-emphasizing filter
'sobel'	Sobel horizontal edge-emphasizing filter
'unsharp'	Unsharp contrast enhancement filter

Filtro Promedio

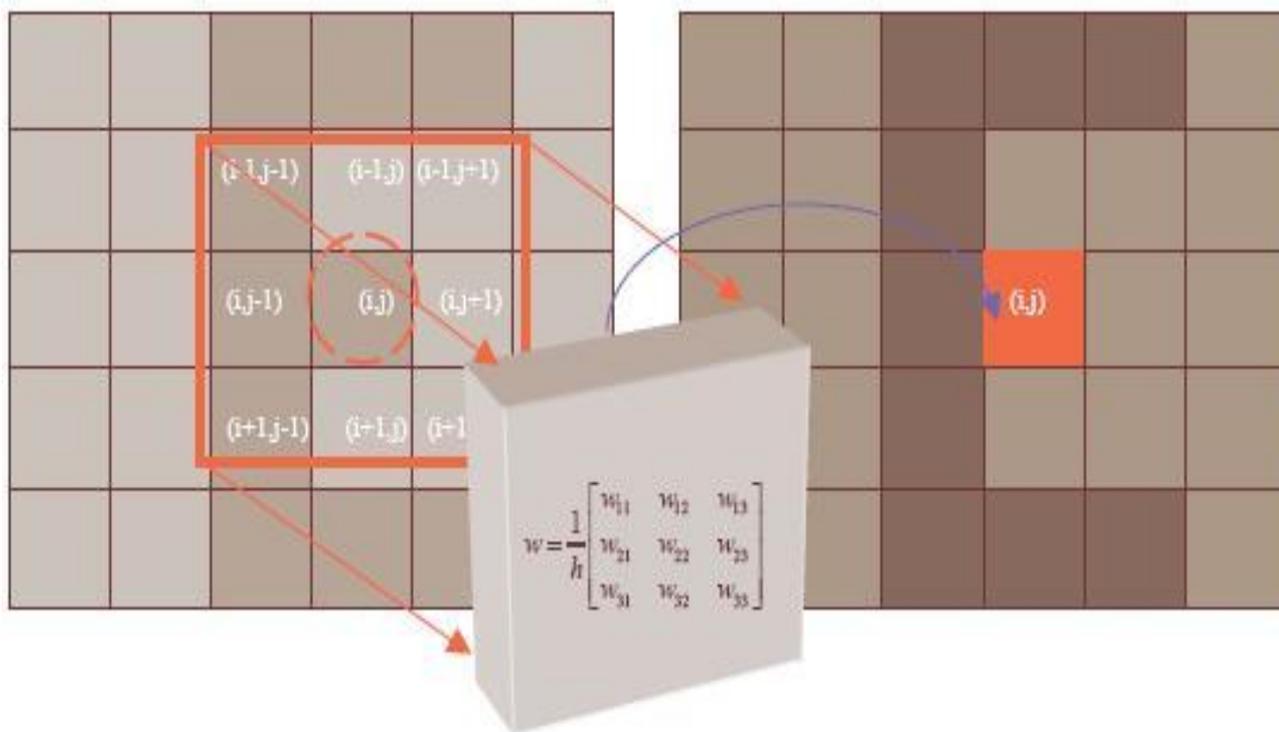
- Obtiene el valor promedio de los pixeles. También se denomina filtro de *media*.

- Su efecto es el difuminado o suavizado de la imagen y se aplica junto con el de mediana para eliminar ruidos.

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

- Este filtro se puede implementar con la siguiente máscara(kernel) para un tamaño 3x3:

Filtro Promedio



Ej. Filtro Promedio



Filtro Gaussiano

- Se usa para suavizar la imagen
- El suavizado es dependiente de la desviación estándar de la mascara.
- A mayor σ^2 el suavizado es mayor.
- La mascara es no lineal pero el filtro es lineal.

Ej. Filtro Gaussiano

$\sigma = 1$



$\sigma = 2$



$\sigma = 3$



$\sigma = 4$



Filtro Unsharp para acentuar contraste

```
>> y=fspecial('unsharp');
```

Imagen Original



Resultado del Filtro



Detectores de Bordes

- Los bordes de una imagen digital se definen como transiciones entre dos regiones de niveles de gris significativamente distintos.
- Métodos basados en el gradiente: detectan los bordes en base a las derivadas espaciales de la imagen que se calculan mediante operadores de convolución
- En Matlab primero se transforma a la imagen original $I(x, y)$ a escala de grises por medio de la función *rgb2gray* para poder aplicar la detección de bordes.

Detección

- La suavización de la imagen evita que se sobredetecten los bordes.
- Los máximos de la primera derivada o los cruces por cero de la segunda derivada permiten detectar los bordes.

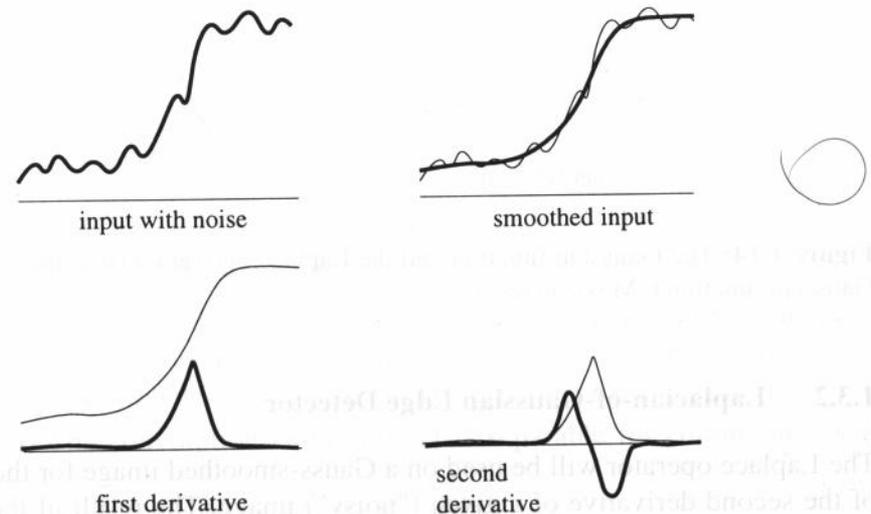
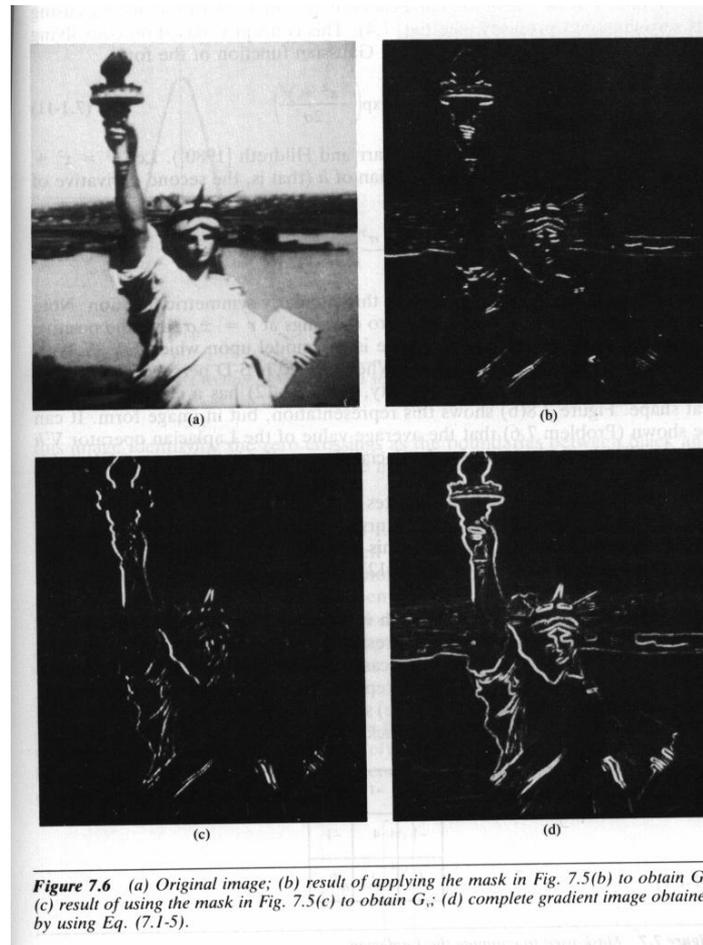


Figure 1.13: Representation of different one-dimensional curves (visualizing gray value distributions in a single image row or column).

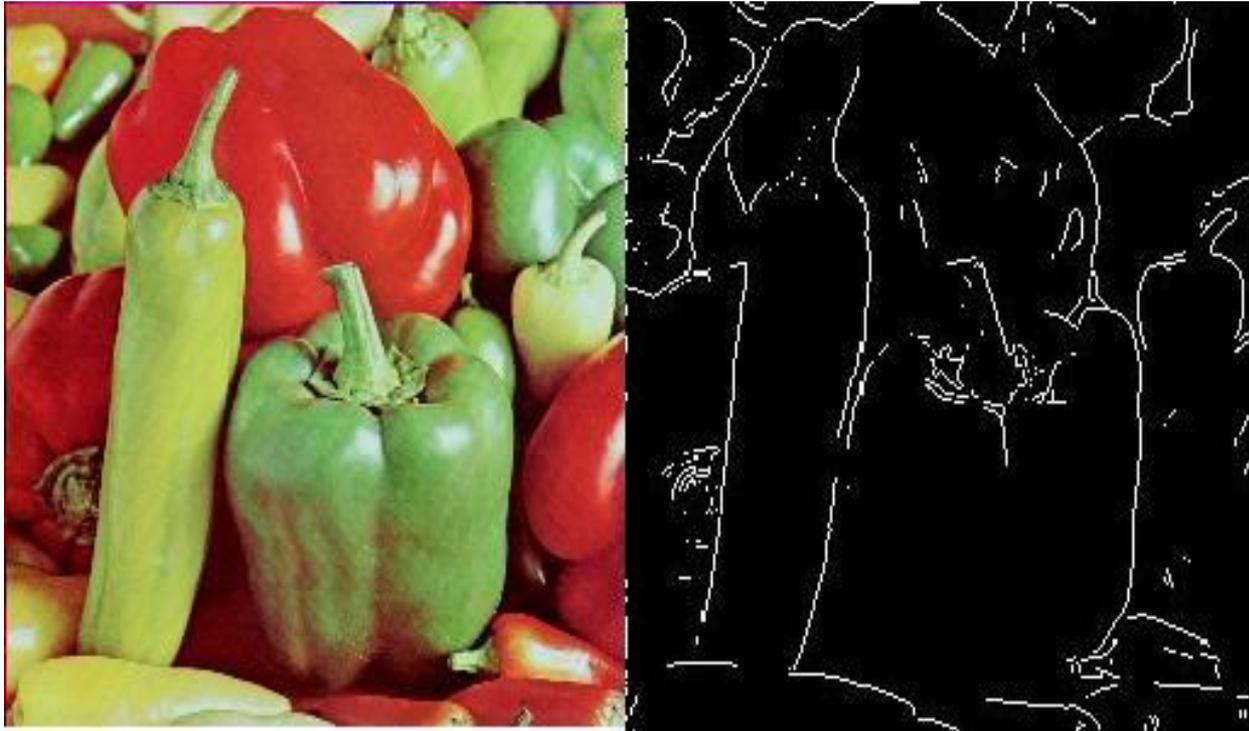
Ej. Detección de borde



Detector Prewitt

- Se define como la aplicación de 8 matrices pixel a pixel a la imagen. La respuesta es la suma de los bordes bien marcados.
- Los nombres de cada matriz se define como un punto cardinal: Norte, Sur, Este, Oeste, Noroeste, Noreste, Suroeste, Sureste.
- Se invoca a la función *edge* para el detector de borde '*prewitt*' asumiendo los parámetros de umbral (*threshold*) y la dirección del gradiente (*direction*) por default. Al definirlo de esta manera se considera en horizontal y vertical.

Método de Prewitt



Prewitt

Prewitt: Acentuar transiciones horizontales

Máscara: $w =$

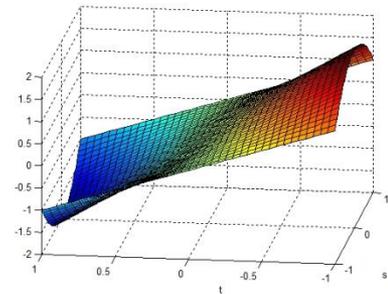
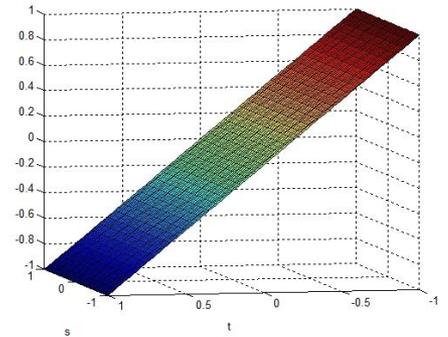
$$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

Sobel: Acentuar transiciones horizontales

Máscara: $w =$

$$\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

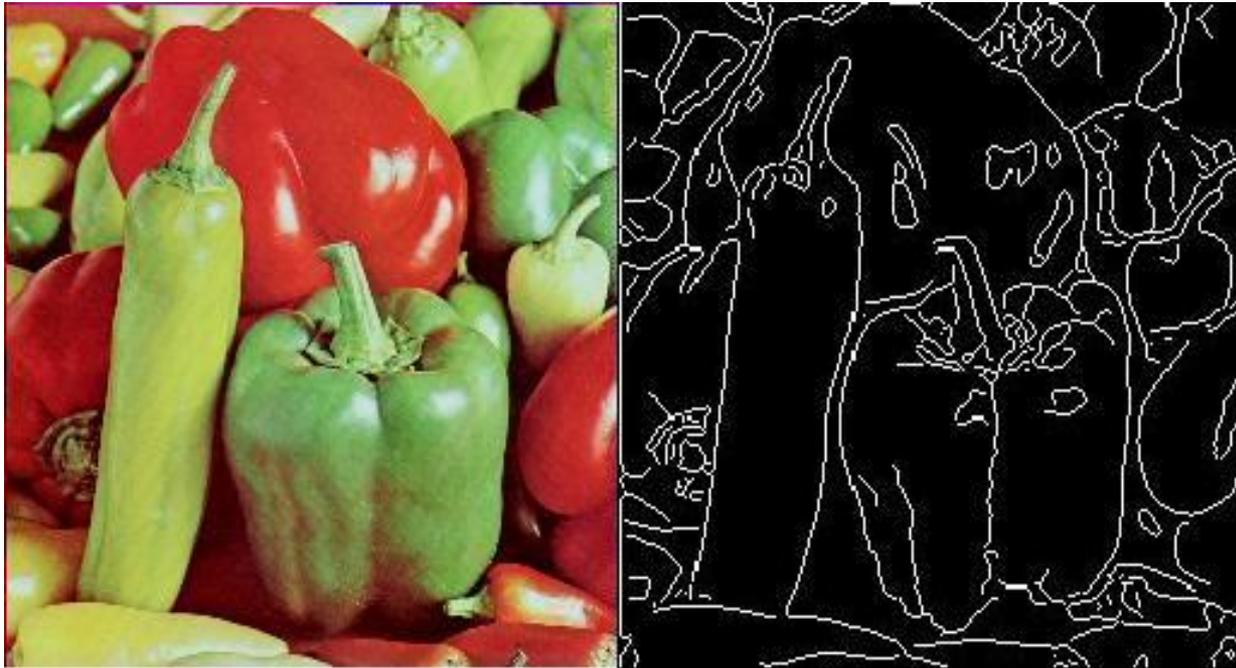
Para acentuar transiciones verticales usar la transpuesta



Método de Canny

- El método utiliza dos umbrales, para detectar los bordes fuertes y débiles, e incluye los bordes débiles en la salida sólo si están conectados a los bordes fuertes.
- En este método se tiene más probabilidades de detectar ciertos bordes débiles y se lo considera como uno de los mejores en la detección de bordes.
- El método de Canny encuentra bordes buscando máximos locales del gradiente.

Método de Canny



Prewitt vs. Canny



Filtros Fotográficos

- Los filtros fotográficos permiten ver las imágenes tal y como desea tomarlas y es la razón por la cual son tan importantes.
- Una de las formas mas fáciles de mejorar su fotografía digital es aprender a utilizar los filtros fotográficos.



Filtros Fotográficos

- **Filtros Polarizadores** .- Es un filtro indispensable para eliminar los brillos en el agua y los reflejos en los vidrios.
- **Filtros de Densidad Neutral (ND)**.- Estos son los filtros necesarios para regular el contraste de las escenas.
- **Filtros de Color**.- Los filtros de color permiten cambiar la tonalidad de sus fotografías.
- **Filtros Especializados**.- Dentro de esta categoría se encuentran los filtros que se sales de las otras clasificaciones.

Tipos de Ruido

Aplicados a la herramienta de Matlab

Aplicación en Matlab

- Para agregar ruido a la imagen de ingreso en Matlab se considera a la función *imnoise*.
- Se detallan los siguientes tipos de Ruido:
 - Ruido Gaussiano
 - Ruido Poisson
 - Ruido Sal & Pimienta (Salt&Pepper)
 - Ruido Speckle

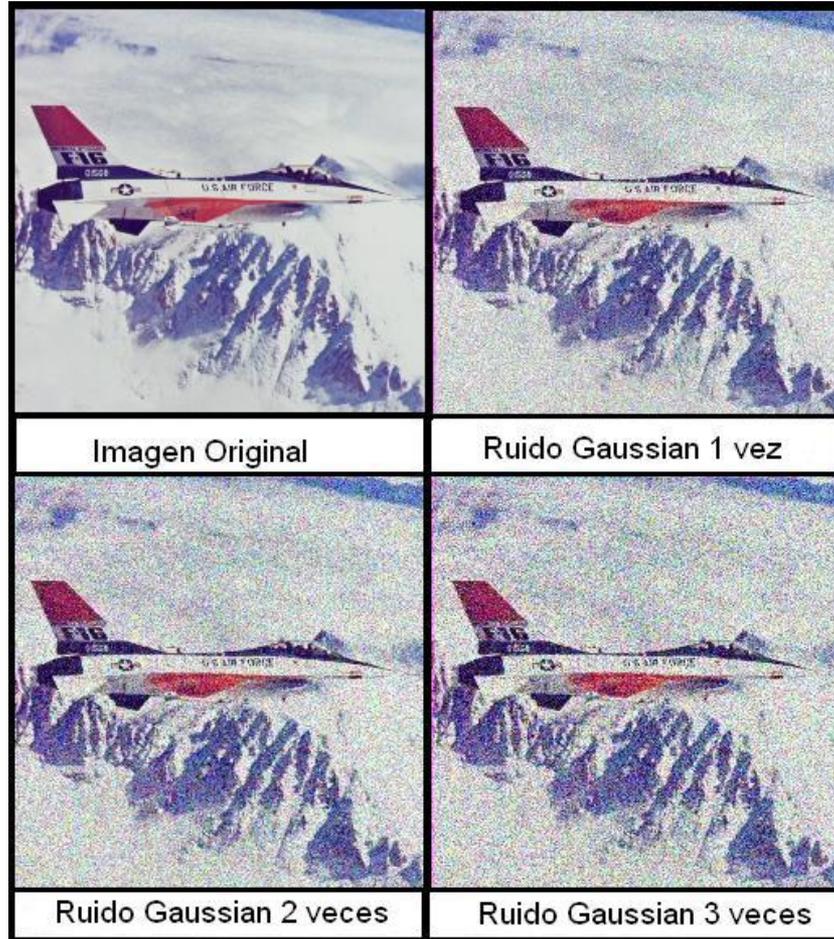
Ruido Gaussiano

- En la herramienta didáctica se asume para la función de ruido Gaussiano que el valor predeterminado es cero ruido media $m(\mu)$ y con una v (de valor $0,01$ de varianza, donde I es la imagen a la cual se va a añadir el ruido.
- Se define ruido blanco como un proceso estocástico que presenta media nula, varianza constante y covarianza nula y si además la distribución es normal, se denomina Ruido Blanco Gaussiano.

$J = imnoise(I, 'gaussian', m, v);$

- Los comandos que se usan para invocar a la función son:

Ej. Ruido Gaussiano

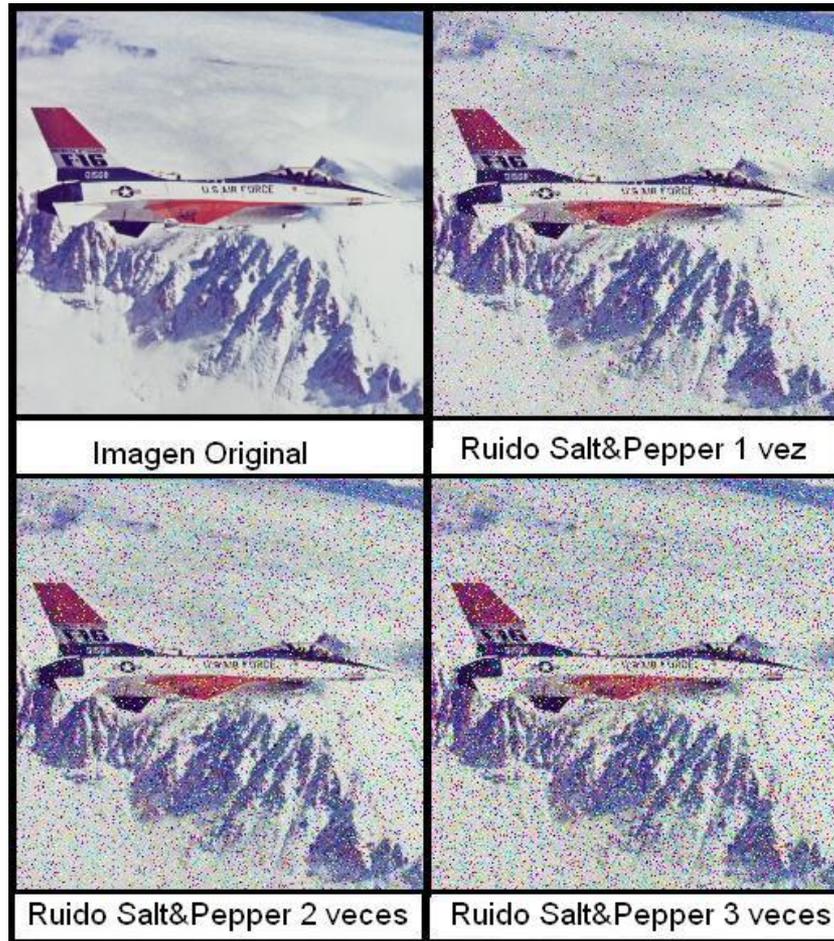


Ruido Sal y Pimienta

- Se define como ocurrencias aleatorias de pixeles completamente blancos y completamente negros. Añade el ruido a la imagen donde d es la densidad del ruido.
- Se afecta aproximadamente a los $d \cdot \text{num}(I)$ pixeles. Y en la función se define por defecto un valor de $d=0,05$.

$J = \text{imnoise}(I, 'salt\&pepper', d);$

Ej. Ruido Sal y pimienta



Ruido Speckle

- Se añade el ruido a la multiplicación de la imagen por medio de la siguiente ecuación:

$$J = I + n * I$$

- Donde n es de distribución uniforme de ruido aleatorio con media 0 y con varianza v . El valor determinado de $v = 0,04$. Estos valores se definen por defecto en Matlab.
- La sintaxis para invocar a la función es:

$$J = \text{imnoise}(I, 'speckle', v);$$

Ej. Ruido Speckle

