



Procesamiento de Imágenes I



Clase Práctica: Detección de Bordos

Procesamiento de Imágenes I

La materia en la carrera

- Materia Optativa no obligatoria.
- Dentro del área de Procesamiento de Señales.
- Correlativas: Análisis y Diseño de Algoritmos II, Arquitectura de Computadoras I.
- A partir de 3er año de Ingeniería de Sistemas.

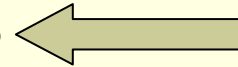
Procesamiento de Imágenes I

Guía de Trabajos Prácticos

Práctico 1: CAPTURA, ALMACENAMIENTO Y REPRESENTACION DE IMAGENES

Práctico 2: REALCE DE IMAGENES

Práctico 3: SEGMENTACIÓN - DETECCIÓN DE BORDES



Práctico 4: OPERACIONES GEOMÉTRICAS Y ALMACENAMIENTO DE OBJETOS

Práctico 5: OPERADORES MORFOLOGICOS

Práctico 6: MODELOS DE COLOR Y FORMATOS DE ARCHIVO

Práctico 7: PATTERN MATCHING

Procesamiento de Imágenes I

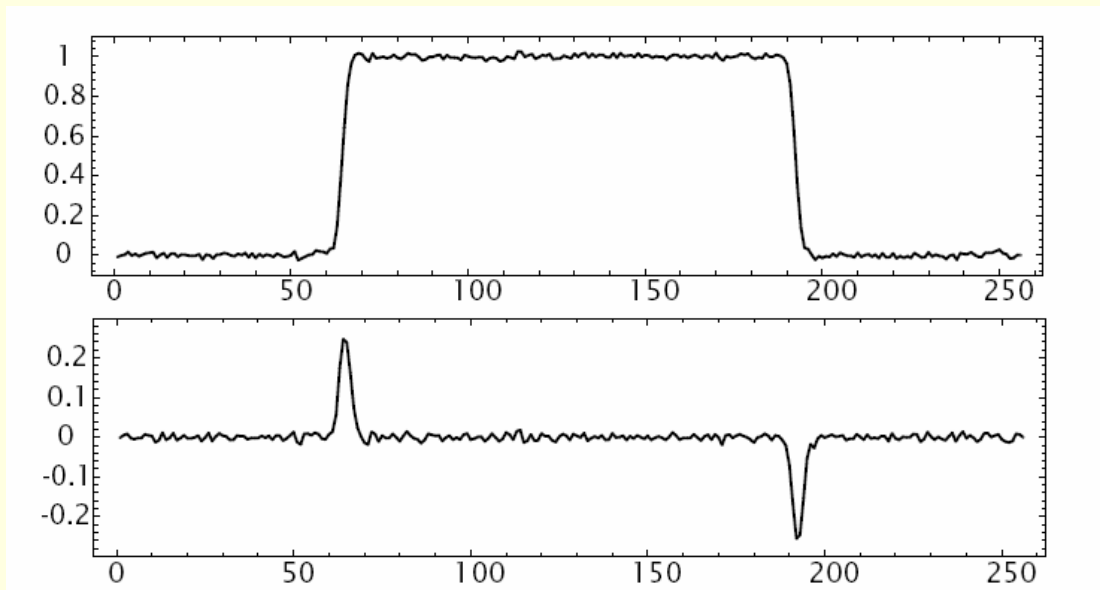
Objetivo de la Clase:

- Implementar los algoritmos de detección de bordes vistos en la teoría.
- Aplicar cada una de ellos a diferentes imágenes.
- Comparar la aplicabilidad de uno u otro en diferentes casos.

Procesamiento de Imágenes I

Borde

Es la discontinuidad del nivel de gris de un píxel respecto de sus vecinos en una imagen digital



Imagen

Derivada 1era

Algo de segmentacion?

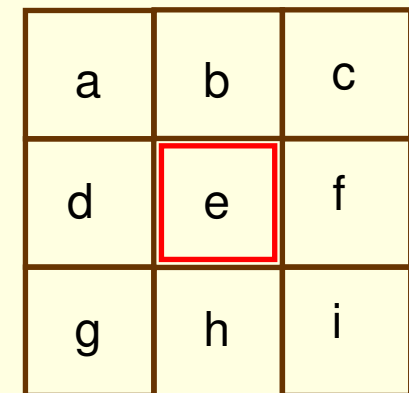
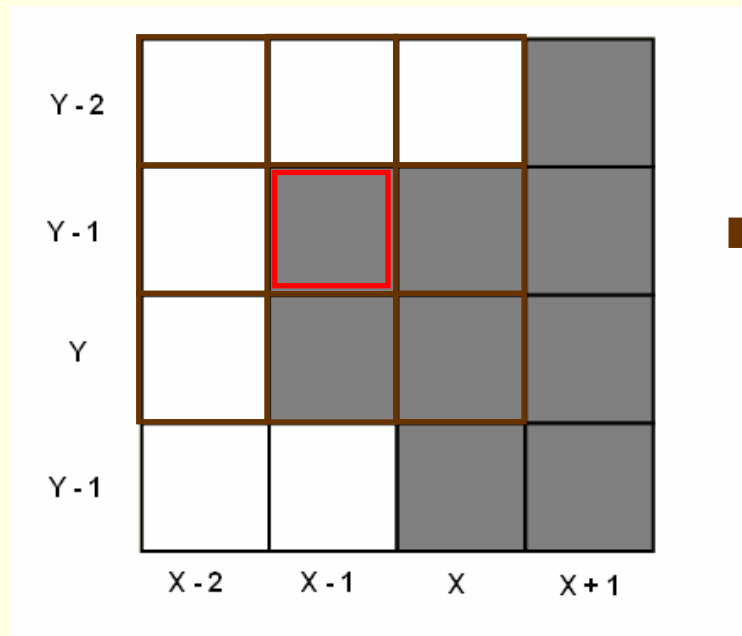
Las técnicas de detección de bordes es una forma de segmentación tambien Filtro pasa alto, pasa bajo Umbrales

Procesamiento de Imágenes I

Como derivar una imagen discreta?

De que forma determinar si un píxel es borde o no tratando de eliminar puntos espureos de la mejor forma ?

Para determinar si un píxel es borde o no, debe compararse con todos sus vecinos



Máscara de Convolución

Procesamiento de Imágenes I

a	b	c
d	e	f
g	h	i

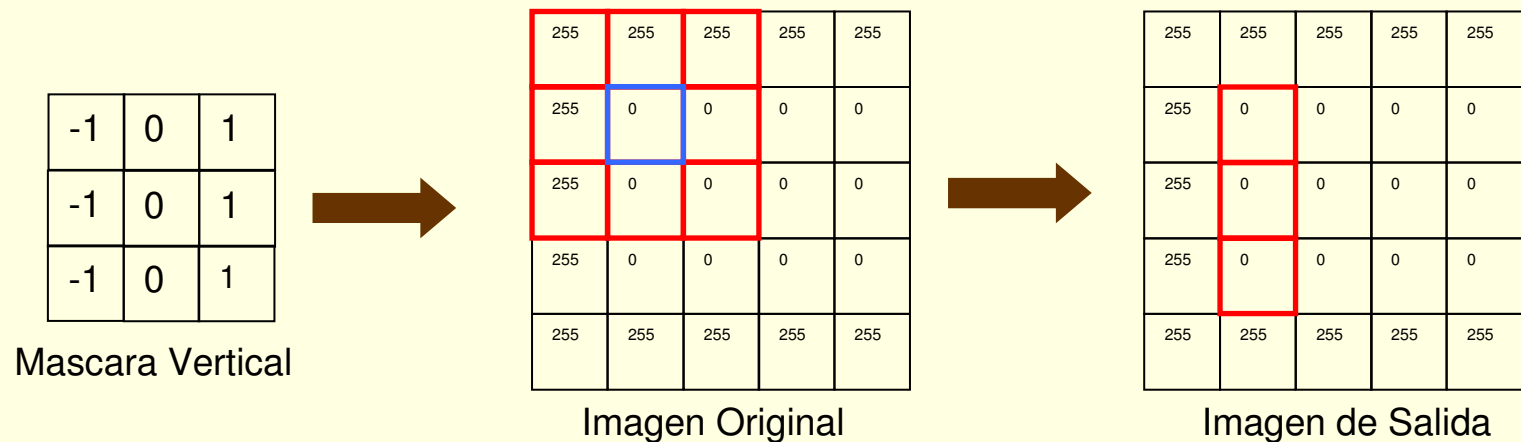
$$O(X, Y) = a * I(x-1, y-1) + b * I(x-1, y) + c * I(x-1, Y-1) + \\ d * I(x, y-1) + e * I(x, y) + f * I(x, y+1) + \\ g * I(x+1, y-1) + h * I(x+1, y) + i * I(x+1, y+1)$$

Cada píxel de la imagen de entrada es evaluado con todos sus vecinos, produciendo un valor de píxel de la imagen de salida.

Los valores de los coeficientes dependen de que característica se quiere resaltar en la dirección que se quiera resaltar.

Procesamiento de Imágenes I

Ejemplo de cómo aplicar una máscara.



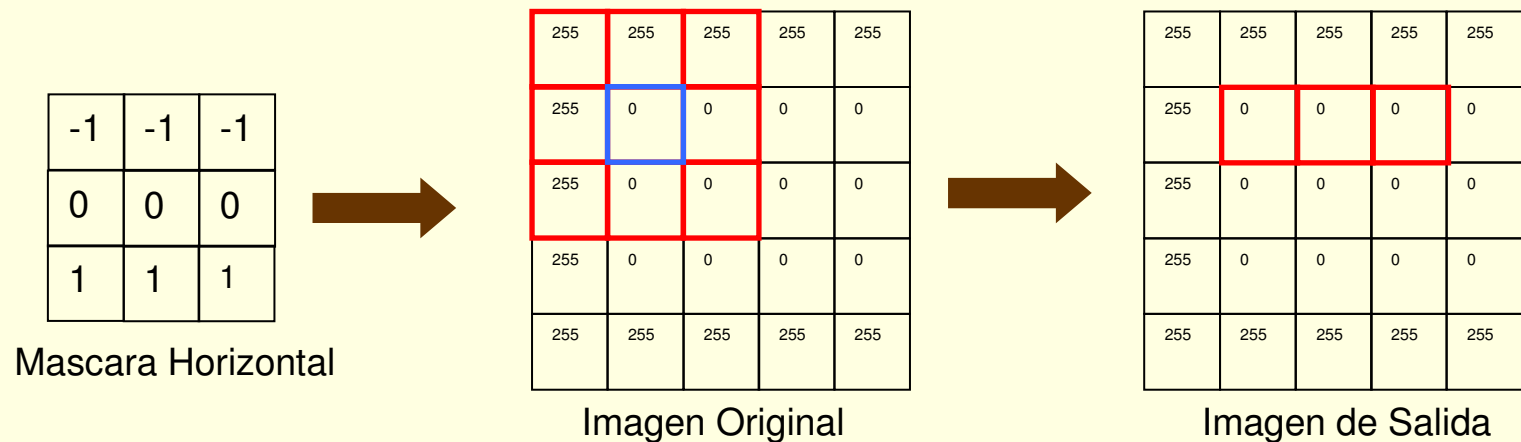
$$O(2,2) = (-255 + 0 + 255) + (-255 + 0 + 0) + (-255 + 0 + 0) = \mathbf{-510}$$

$$O(2,3) = (-255 + 0 + 255) + (-0 + 0 + 0) + (-0 + 0 + 0) = \mathbf{0}$$

$$O(3,2) = (-255 + 0 + 255) + (-255 + 0 + 0) + (-255 + 0 + 0) = \mathbf{-510}$$

Procesamiento de Imágenes I

Ejemplo de cómo aplicar una máscara.



$$O(2,2) = (-255 - 255 - 255) + (0 + 0 + 0) + (255 + 0 + 0) = \mathbf{-510}$$

$$O(2,3) = (-255 - 255 - 255) + (0 + 0 + 0) + (0 + 0 + 0) = \mathbf{-765}$$

$$O(3,2) = (-255 + 0 + 0) + (0 + 0 + 0) + (255 + 0 + 0) = \mathbf{0}$$

Procesamiento de Imágenes I

Técnicas de detección de Bordes

Dada una imagen original, las técnicas de detección de bordes dan como resultado otra imagen en la cual solo se resaltan los bordes presentes en la original.



Procesamiento de Imágenes I

Pasos generales de las técnicas de Detección de Bordes:

1. Aplicar Mascaras
2. Sumar los resultados de las mascararas
3. Categorizar las intensidades
4. Normalizar el resultado (llevar el mínimo a 0 y el máximo a 255)

Las técnicas mas conocidas son: Sobel, Prewitt, Roberts, Laplace

Procesamiento de Imágenes I

Ejercicio 4. Cargar una imagen y aplicarle el filtro de Sobel

```
Procedure Sobel() {  
  for j:= 1 to Imagen1.Height-1 do  
    for i := 1 to Imagen1.Width -1 do {  
      ImagenH (i, j) := OHoriz(Imagen1, i, j);  
      ImagenV (i, j) := OVert(Imagen1, i, j);  
    }  
  ImagenOut := Suma (ImagenH, ImagenV)  
}
```

1. Imagen1 es la imagen original.
2. ImagenH contiene el resultado de aplicar la mascara Horizontal a Imagen1.
3. ImagenV contiene el resultado de aplicar la mascara Vertical a Imagen 1.
4. ImagenOut contiene el resultado.

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

Mascara Vertical

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

Mascara Horizontal

Procesamiento de Imágenes I

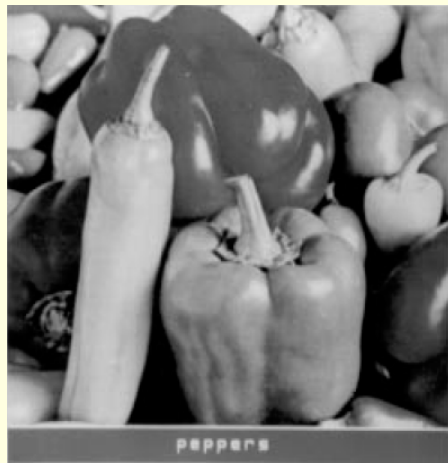


Imagen Original



Resultado: Mascara Horizontal



Resultado: Mascara Vertical



Imagen Resultante

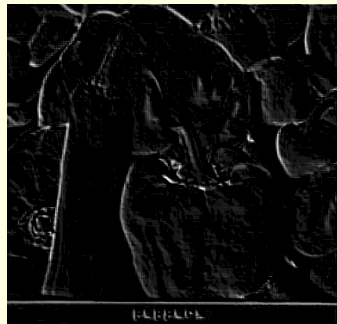
Procesamiento de Imágenes I

Ejercicio 5. Cargar una imagen y aplicarle el filtro de Prewitt

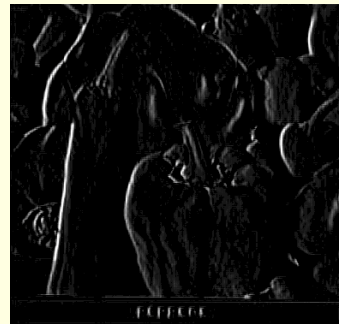
$$\begin{array}{cccc} \begin{bmatrix} 1 & 1 & -1 \\ 1 & -2 & -1 \\ 1 & 1 & -1 \end{bmatrix} & \begin{bmatrix} 1 & -1 & -1 \\ 1 & -2 & -1 \\ 1 & 1 & 1 \end{bmatrix} & \begin{bmatrix} -1 & -1 & -1 \\ 1 & -2 & 1 \\ 1 & 1 & 1 \end{bmatrix} & \begin{bmatrix} -1 & -1 & 1 \\ -1 & -2 & 1 \\ 1 & 1 & 1 \end{bmatrix} \\ \text{East} & \text{Northeast} & \text{North} & \text{Northwest} \\ \mathbf{H_1} & \mathbf{H_2} & \mathbf{H_3} & \mathbf{H_4} \\ \\ \begin{bmatrix} -1 & 1 & 1 \\ -1 & -2 & 1 \\ -1 & 1 & 1 \end{bmatrix} & \begin{bmatrix} 1 & 1 & 1 \\ -1 & -2 & 1 \\ -1 & -1 & 1 \end{bmatrix} & \begin{bmatrix} 1 & 1 & 1 \\ 1 & -2 & 1 \\ -1 & -1 & -1 \end{bmatrix} & \begin{bmatrix} 1 & 1 & 1 \\ 1 & -2 & -1 \\ 1 & -1 & -1 \end{bmatrix} \\ \text{West} & \text{Southwest} & \text{South} & \text{Southeast} \\ \mathbf{H_5} & \mathbf{H_6} & \mathbf{H_7} & \mathbf{H_8} \end{array}$$

```
Procedure prewitt(){  
  
  for j:= 1 to Imagen1.Height-1 do  
    for i := 1 to Imagen1.Width -1 do {  
      Calcular las 8 imagenes (todas las direcciones)  
    }  
  ImagenOut := Suma (Imagen1,.. Imagen8)  
}
```

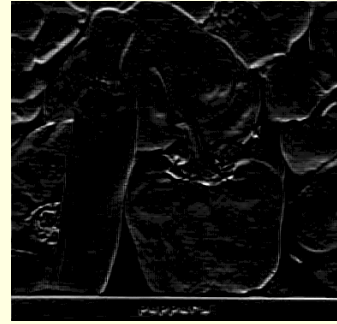
Procesamiento de Imágenes I



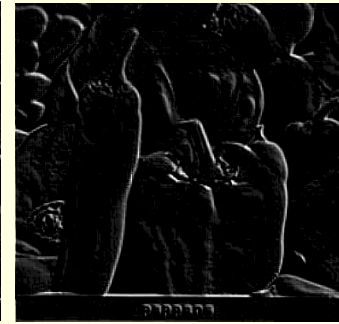
Norte



Noreste



Sur



Suroeste



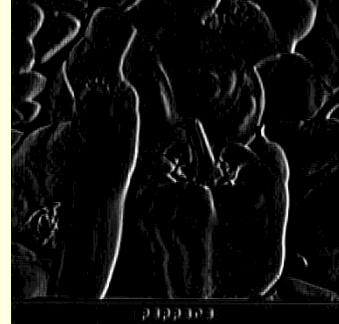
Noroeste



Oeste



Sureste



Este



Imagen Resultante

Procesamiento de Imágenes I

Comparación de resultados



Sobel



Prewitt