

Trabajo Práctico 1

Procesamiento de imágenes para la detección de bordes

Organización del Computador 2

2do. Cuatrimestre 2007

1. Introducción teórica

La detección de bordes en una imagen reduce significativamente la cantidad de información y filtra la información innecesaria, preservando las características fundamentales de la imagen. De allí que los métodos para detectar bordes sean una herramienta muy poderosa para el procesamiento de imágenes. Podemos definir como un borde a los pixels donde la intensidad de la imagen cambia de forma abrupta. Si consideramos una función de intensidad de la imagen, entonces lo que buscamos son saltos en dicha función.

La idea básica detrás de cualquier detector de bordes es el cálculo de un operador local de derivación. Para simplificar, estudiaremos el caso de imágenes en escala de grises. En la figura 1 se puede ver este concepto. En la parte izquierda se puede ver una imagen de una banda clara sobre un fondo oscuro, el perfil a lo largo de una línea horizontal y la primera y segunda derivada de dicho perfil. Se puede observar que el perfil del borde se ha modelado como una discontinuidad suave. Esto tiene en cuenta el hecho de que en las imágenes reales los bordes están ligeramente desenfocados.

Como se puede observar en la figura 1 la primera derivada es positiva para cambio a nivel de gris más claro, negativa en caso contrario y cero en aquellas zonas con nivel de gris uniforme. La segunda derivada presenta valor positivo en la zona oscura de cada borde, valor negativo en la zona clara de cada borde y valor cero en las zonas de valor de gris constante y justo en la posición de los bordes. El valor de la magnitud de la primera derivada nos sirve para detectar la presencia de bordes, mientras que el signo de la segunda derivada nos indica si el pixel pertenece a la zona clara o a la zona oscura. Además, la segunda derivada presenta siempre un cero en el punto medio de la transición. Esto puede ser muy útil para localizar bordes en una imagen.

Aunque lo que llevamos dicho hasta aquí se refiere a perfiles unidimensionales, la extensión a dos dimensiones es inmediata. Simplemente se define el perfil en la dirección perpendicular a la dirección del borde y la interpretación anterior seguirá siendo válida. La primera derivada en cualquier punto de la imagen vendrá dada por la magnitud del gradiente, mientras que la segunda derivada vendrá dada por el operador Laplaciano. La mayoría de los métodos de detección de bordes pueden dividirse en dos grandes grupos.

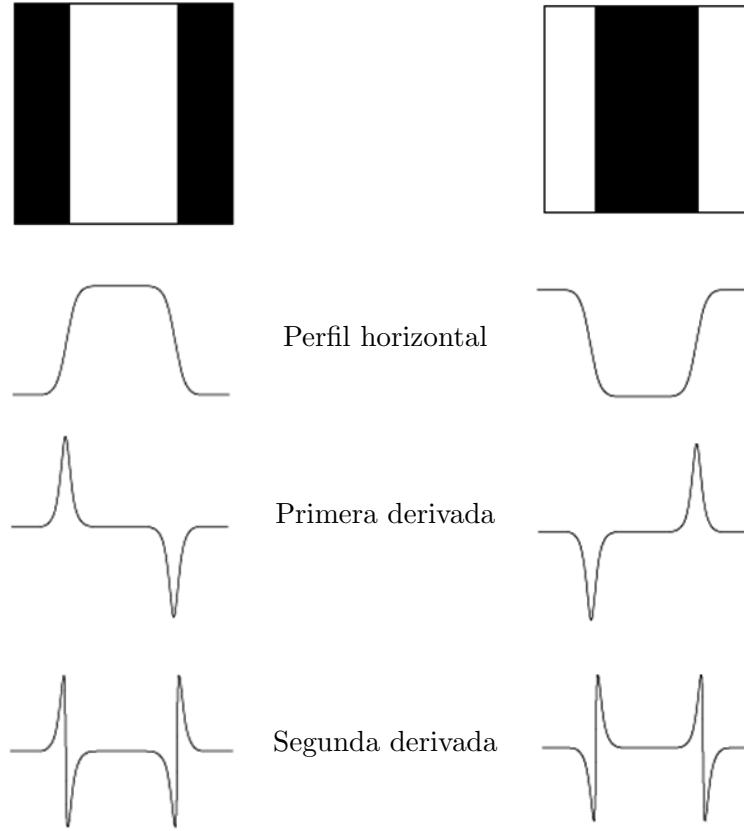


Figura 1: Detección de bordes empleando operadores de derivación. La primera derivada es positiva para cambio a nivel de gris más claro, negativa en caso contrario y cero en zonas con nivel de gris uniforme. La segunda derivada tiene un cero en la posición de cada borde.

El gradiente de una imagen $I(x, y)$ en la posición (x, y) viene dado por el vector:

$$\nabla I = \begin{bmatrix} I_x \\ I_y \end{bmatrix} = \begin{bmatrix} \frac{\partial I}{\partial x} \\ \frac{\partial I}{\partial y} \end{bmatrix}. \quad (1)$$

El vector gradiente siempre apunta en la dirección de la máxima variación de la imagen I en el punto (x, y) . Para los métodos de gradiente nos interesa conocer la magnitud de este vector, denominado simplemente como gradiente de la imagen, denotado por $\|\nabla I\|$ y dado por:

$$\|\nabla I\| = \sqrt{I_x^2 + I_y^2}. \quad (2)$$

Esta cantidad representa la variación de la imagen $I(x, y)$ por unidad de distancia en la dirección del vector ∇I . En general, el gradiente de la imagen se suele aproximar mediante la expresión

$$||\nabla I|| \approx |I_x| + |I_y|, \quad (3)$$

que es mucho más simple de implementar computacionalmente.

Para el cálculo del gradiente de la imagen en cada píxel tenemos que obtener las derivadas parciales. Las derivadas se pueden implementar digitalmente de varias formas. Una forma es mediante máscaras de tamaño 3×3 o incluso más grandes. La ventaja de utilizar máscaras grandes es que los errores producidos por efectos del ruido son reducidos mediante promedios locales tomados en los puntos en donde se superpone la máscara. Por otro lado, las máscaras normalmente tienen tamaños impares, de forma que los operadores se encuentran centrados sobre el píxel en el que se calcula el gradiente. El requisito básico de un operador de gradiente es que la suma de los coeficientes de la máscara sea nula, para que la derivada de una zona uniforme de la imagen sea cero.

Los operadores de gradiente común (o gradiente ortogonal) encuentran bordes horizontales y verticales. Estos operadores trabajan mediante convolución, es decir, se aplica una máscara sucesivamente sobre cada píxel de la imagen. Los operadores de Roberts, Prewitt, Sobel y Frei-Chen son operadores dobles o de dos etapas. La detección de bordes se realiza en dos pasos. En el primero se buscan bordes horizontales utilizando la máscara que corresponde a la derivada parcial en x . En el segundo paso se buscan los bordes verticales utilizando la máscara que corresponde a la derivada parcial en y . Finalmente, se suman ambos para obtener el gradiente de la imagen en cada píxel. En la figura 2 se pueden ver los operadores de Roberts, Prewitt, Sobel y Frei-Chen para determinar bordes horizontales (izquierda) y verticales (derecha).

Los operadores de Sobel y de Frei-Chen tienen la ventaja de que proporcionan un suavizado además del efecto de derivación. Ya que la derivación acentúa el ruido, el efecto de suavizado es particularmente interesante, puesto que elimina parte del ruido.

Una vez que aplicamos el operador de gradiente, tenemos que utilizar un método de umbralización para determinar cuáles píxeles son considerados parte de un borde. La forma más sencilla es fijar un umbral y considerar borde a todo píxel cuya intensidad supere dicho umbral. Si fijamos un umbral bajo, más bordes van a ser detectados, pero el resultado se vuelve crecientemente más susceptible al ruido y a seleccionar características irrelevantes de la imagen. Contrariamente, un umbral alto puede que no detecte bordes suaves o que se van atenuando.

Una forma de tratar este problema es mediante la umbralización con histéresis. Este método usa múltiples umbrales. Se comienza utilizando un umbral superior para encontrar el inicio de un borde. Una vez que tenemos este punto de partida, se recorre la línea de borde marcando cada píxel adyacente al anterior recorrido cuya intensidad está por encima del umbral inferior. Dejamos de marcar el borde sólo cuando no hay píxeles adyacentes con intensidad superior al umbral inferior. De esta forma podemos seguir bordes que se atenúan, sin tener que marcar el ruido de la imagen como bordes.

1	0
0	-1

0	1
-1	0

(a)

-1	0	1
-1	0	1
-1	0	1

-1	-1	-1
0	0	0
1	1	1

(b)

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

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

(c)

-1	0	1
$-\sqrt{2}$	0	$\sqrt{2}$
-1	0	1

-1	$-\sqrt{2}$	-1
0	0	0
1	$\sqrt{2}$	1

(d)

Figura 2: Operadores de derivación: (a) de Roberts, (b) de Prewitt, (c) de Sobel y (d) de Frei-Chen.

Por último, dijimos que íbamos a simplificar el problema tratando sólo con imágenes en escala de grises. Para eso, debemos contar con una función que sea capaz de *monocromatizar* una imagen color. La función más sencilla para hacer esto es:

$$f(x, y) = \sqrt[\epsilon]{\alpha R^\epsilon + \beta G^\epsilon + \gamma B^\epsilon}, \quad (4)$$

donde la función f se aplica a cada pixel (x, y) , RGB son sus colores, α , β y γ son coeficientes entre 0 y 1 y ϵ es un exponente entre 1 e ∞ .

2. Enunciado

Hacer un programa para el procesamiento simple de imágenes en archivos BMP. El programa debe permitir monocromatizar una imagen, aplicar los operadores de derivación de Roberts, de Prewitt y de Sobel, para realzar los bordes y umbralizar con histéresis para devolver una imagen que sólo contenga los bordes de la imagen original. Debe estar programado de manera modular, para que sea fácil agregar otros tipos de procesamiento en el futuro.

Para esta primera versión analizaremos sólo los casos extremos de ϵ en la función de monocromatizar ($\epsilon = 1$ y $\epsilon = \infty$) y fijaremos $\alpha = \beta = \gamma = 1/3$.

Entonces nos quedan dos posibles funciones para monocromatizar:

- $f_1(x, y) = 1/3(R + G + B)$ ($\epsilon = 1$)
- $f_2(x, y) = \max(R, G, B)$ ($\epsilon = \infty$)

Se puede escribir la parte de interacción con el usuario y manejo de archivos en lenguaje C. Las funciones de procesamiento de imágenes deben estar escritas en lenguaje ensamblador. El programa debe recibir en la línea de comandos el nombre del archivo de entrada, el nombre del archivo de salida y la operación:

- r1 = realzar bordes con el operador de Roberts.
- r2 = realzar bordes con el operador de Prewitt.
- r3 = realzar bordes con el operador de Sobel.
- m1 = monocromatizar con $\epsilon = 1$.
- m2 = monocromatizar con $\epsilon = \infty$
- u = umbralizar (en este caso también toma dos enteros para los umbrales máximo y mínimo)

Otras especificaciones:

- Soporta formato BMP de color real de 24 bits (no es necesario soportar el resto).
- El ancho máximo de las imágenes es de 10000 pixels.
- El tamaño máximo del archivo es de 500 Mbytes.

Informe

Debe reflejar el trabajo hecho para obtener el resultado, las decisiones tomadas (con el estudio de sus alternativas), las estructuras de datos usadas (con gráficos y/o dibujos si ayudan a clarificar), las pruebas que hayan hecho para tomar decisiones o al final para buscar errores en el producto final, etcétera. Debe contar como mínimo con los siguientes capítulos: introducción, desarrollo, discusión y conclusiones. Estar estructurado top-down o sea leyendo la introducción se debe saber qué se hizo y cuáles son las partes más importantes. Después de leer los primeros capítulos se debe saber cada cosa que se hizo y como se hicieron las más importantes. En el resto de los capítulos se debe poder leer el detalle de todo lo hecho.

Además, el informe debe incluir:

- Carátula con número del grupo y los nombres de los integrantes con número de libreta y email
- Manual del usuario
- Algunas imágenes con las que hayan probado y su correspondiente procesamiento.
- Instrucciones para el corrector, por ejemplo como ensamblar los archivos fuente para obtener el ejecutable.
- Lista de todos los archivos entregados.

Entrega

La fecha de entrega de este trabajo es martes 16 de octubre, en el horario de clase (de 17 a 22 hs). No se aceptarán trabajos pasada esa fecha. Para poder contar con una instancia de recuperación del trabajo, es necesario haber entregado una primera versión en la fecha antes mencionada.

La entrega se realizará en un CD que debe incluir, los ejecutables, todos los archivos fuentes necesarios para crearlos, imágenes de prueba con su correspondiente procesamiento y el informe en pdf. Si desarrollaron prototipos en lenguaje C para resolver el trabajo primero en alto nivel, deben entregarlos e incluir los resultados obtenidos en el informe.