Filtros en el dominio espacial

Lección 05.2

Dr. Pablo Alvarado Moya

MP6123 Procesamiento Digital de Imágenes
Programa de Maestría en Electrónica
Énfasis en Procesamiento Digital de Señales
Escuela de Ingeniería Electrónica
Tecnológico de Costa Rica

II Cuatrimestre 2012

Contenido

Filtros lineales

- Suavizamiento (Filtros paso-bajo)
- Aproximación de derivación (Filtros paso-alto)
- Laplaciano (Filtros paso-alto)
- Nitidez

Filtros de media móvil

$$h(x,y) = \frac{1}{N^2} \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & \dots & 1 \end{bmatrix} = \frac{1}{N^2} \begin{bmatrix} 1 & 1 & \dots & 1 \\ 1 & 1 & \dots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \dots & 1 \end{bmatrix}$$

Filtros gaussianos y binomiales

Gaussiano:

$$g(x, y, \sigma) = g(x, \sigma)g(y, \sigma)$$
$$g(x, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{1}{2}\left(\frac{x}{\sigma}\right)^2}$$

Binomial:

$$B_N(x, y) = B_N(x)B_N(y)$$

$$B_N(x) = \frac{1}{K}\binom{N}{x} \qquad \binom{N}{x} = \frac{N!}{(N-n)!N!} \quad K = \sum_{x=0}^{N}\binom{N}{x}$$

Triángulo de Pascal

$$
\begin{array}{ccccccccccccccccc}
 & & & & & & & & 1 & & & & & & & & \\
 & & & & & & & 1 & & 1 & & & & & & & \\
 & & & & & & 1 & & 2 & & 1 & & & & & & \\
 & & & & & 1 & & 3 & & 3 & & 1 & & & & & \\
 & & & & 1 & & 4 & & 6 & & 4 & & 1 & & & & \\
 & & & 1 & & 5 & & 10 & & 10 & & 5 & & 1 & & & \\
 & & 1 & & 6 & & 15 & & 20 & & 15 & & 6 & & 1 & & \\
 & 1 & & 7 & & 21 & & 35 & & 35 & & 21 & & 7 & & 1 & \\
1 & & 8 & & 28 & & 56 & & 70 & & 56 & & 28 & & 8 & & 1 \\
 & & & & & & & & \vdots & & & & & & & &
\end{array}
$$

$$B_N = \frac{1}{K}\underbrace{[1 \quad 1] * [1 \quad 1] * \cdots * [1 \quad 1]}_{N \text{ veces}}$$

Gradiente

En espacio continuo:

$$\nabla f = \begin{bmatrix} \frac{\partial}{\partial x} f(x,y) \\ \frac{\partial}{\partial y} f(x,y) \end{bmatrix}$$

Con frecuencia se utilizan

$$|\nabla f| = \sqrt{\left(\frac{\partial}{\partial x} f(x,y)\right)^2 + \left(\frac{\partial}{\partial y} f(x,y)\right)^2}$$

$$\angle \nabla f = \arctan\left(\frac{\frac{\partial}{\partial y} f(x,y)}{\frac{\partial}{\partial x} f(x,y)}\right)$$

Gradiente en el espacio discreto (1)

Aproximación por diferencias

Aproximación hacia atrás (Kernel $[\underset{\uparrow}{1} \quad -1]$):

$$\frac{\partial}{\partial x} f(x,y) \approx f(x,y) - f(x-1,y)$$
$$\frac{\partial}{\partial y} f(x,y) \approx f(x,y) - f(x,y-1)$$

Aproximación hacia adelante (Kernel $[1 \quad \underset{\uparrow}{-1}]$):

$$\frac{\partial}{\partial x} f(x,y) \approx f(x+1,y) - f(x,y)$$
$$\frac{\partial}{\partial y} f(x,y) \approx f(x,y+1) - f(x,y)$$

Gradiente en el espacio discreto (2)

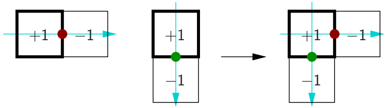
Aproximación por diferencias

Aproximación centrada (Kernel $[1 \quad \underset{\uparrow}{0} \quad -1]$):

$$
\begin{aligned}
\frac{\partial}{\partial x} f(x,y) &\approx \frac{1}{2}(f(x+1,y) - f(x-1,y)) \\
\frac{\partial}{\partial y} f(x,y) &\approx \frac{1}{2}(f(x,y+1) - f(x,y-1))
\end{aligned}
$$

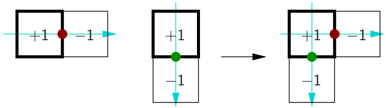
Gradiente en el espacio discreto
Kernel de Roberts

- Aproximaciones por diferencias: componentes desplazadas

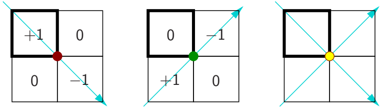


Gradiente en el espacio discreto
Kernel de Roberts

- Aproximaciones por diferencias: componentes desplazadas

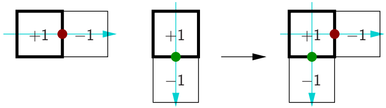


- Aproximación con kernels de Roberts

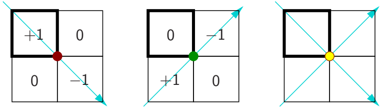


Gradiente en el espacio discreto
Kernel de Roberts

- Aproximaciones por diferencias: componentes desplazadas



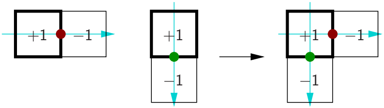
- Aproximación con kernels de Roberts



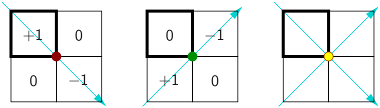
- Derivación amplifica ruido de alta frecuencia

Gradiente en el espacio discreto
Kernel de Roberts

- Aproximaciones por diferencias: componentes desplazadas



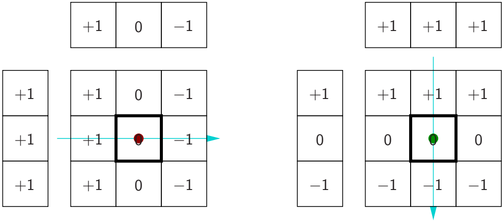
- Aproximación con kernels de Roberts



- Derivación amplifica ruido de alta frecuencia
- Reducción de ruido en dirección ortogonal con filtro pasa-bajos

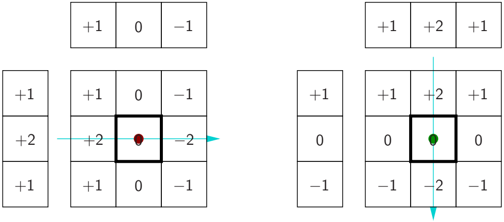
Gradiente en el espacio discreto

Kernel de Prewitt



- Reducción de ruido utilizando filtro de media móvil
- Aproximación central para gradiente
- Factor de normalización de $1/6$
- Origen: aproximación cuadrática de imagen en Vecindad-8

Gradiente en el espacio discreto
Kernel de Sobel



- Reducción de ruido utilizando filtro binomial
- Aproximación central para gradiente
- Factor de normalización de $1/8$
- Origen: aproxima DoG

Otros kernels para gradientes

- Robinson (no separable)

$$
\begin{bmatrix} +1 & +1 & -1 \\ +1 & \boxed{-2} & -1 \\ +1 & +1 & -1 \end{bmatrix}
\qquad
\begin{bmatrix} +1 & +1 & +1 \\ +1 & \boxed{-2} & +1 \\ -1 & -1 & -1 \end{bmatrix}
$$

- Kirsch (no separable, 4 direcciones)

$$
\begin{bmatrix} +3 & +3 & -5 \\ +3 & \boxed{0} & -5 \\ +3 & +3 & -5 \end{bmatrix}
\qquad
\begin{bmatrix} +3 & +3 & +3 \\ +3 & \boxed{0} & +3 \\ -5 & -5 & -5 \end{bmatrix}
$$

- Ando (optimizado para invarianza a rotación)

Notas sobre el gradiente

- En literatura aparecen rotados 180° (inconsistente pero sin mayor efecto) ¿origen?
- Sobel, Robinson y Kirsch: Versiones a $\pm 45^\circ$

DoG

Derivative of Gaussians

- Idea:
  1. Filtrar con kernel gaussiano para reducir ruido
  2. Gradiente/Derivar resultado
- Pero:

$$
\begin{aligned}
o(x,y) &= \nabla(g(x,y,\sigma) * i(x,y)) \\
&= i(x,y) * \nabla g(x,y,\sigma)
\end{aligned}
$$

- Más económico calcular $\nabla g(x,y,\sigma)$:

$$\frac{\partial g}{\partial x} = -\frac{x}{\sigma^2} g(x,y,\sigma) \qquad \frac{\partial g}{\partial y} = -\frac{y}{\sigma^2} g(x,y,\sigma)$$

- Sobel se considera aproximación de DoG

Laplaciano

- Gradiente con dos componentes es **anisotrópico**
  $\Rightarrow$ ambas componentes cambian si imagen rota sobre pixel
- Laplaciano es **isotrópico**
  $\Rightarrow$ único valor no cambia si imagen rota sobre pixel
- Definición en espacio contínuo:

$$\nabla^2 f(x,y) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

Laplaciano discreto

- Definición en espacio discreto basada en aproximación en diferencias:

$$\frac{\partial^2 f}{\partial x^2} = f(x+1,y) - 2f(x,y) + f(x-1,y)$$
$$\frac{\partial^2 f}{\partial y^2} = f(x,y+1) - 2f(x,y) + f(x,y-1)$$

- El laplaciano discreto se implementa con

0	+1	0
+1	$\boxed{-4}$	+1
0	+1	0

- Otras variantes existen para reducir ruido (e.g. LoG)
- Laplaciano no es separable

Usos del Laplaciano

- Medición de "bordicidad" (*edgeness*)
- Mejora de nitidez:

$$g(x, y) = f(x, y) + c\nabla^2 f(x, y)$$

(a imagen se le amplifican los bordes)

Filtros no lineales

Filtros de Rango

Filtros de rango
Rank ordered filters

- También llamados filtros por orden (*order-statistic filters*)
- Son **no lineales** $\Rightarrow$ no aplica convolución
- Se toma el elemento r de los píxeles en la vecindad, ordenados.
- Ejemplos: filtro máximo, filtro mínimo, filtro mediana

Resumen

1. Filtros lineales
   - Máscaras de suavizamiento
   - Aproximación del gradiente
   - Laplaciano
2. Filtros de rango