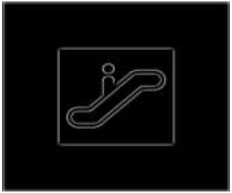


Procesamiento de imágenes digitales

Luis Baumela
Departamento de Inteligencia Artificial
Universidad Politécnica de Madrid

POLITÉCNICA

Índice

Introducción

El procesamiento de imágenes digitales transforma una imagen en otra que:

- tiene una mejor distribución en los niveles de gris (transformaciones puntuales)

Introducción

El procesamiento de imágenes digitales transforma una imagen en otra que:

- tiene una mejor distribución en los niveles de gris (transformaciones puntuales)
- tiene menos ruido (filtrado paso bajo)

Introducción

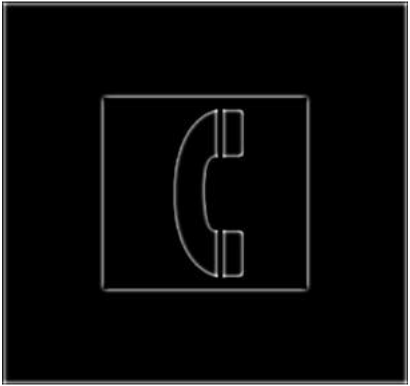
El procesamiento de imágenes digitales transforma una imagen en otra que:

- tiene una mejor distribución en los niveles de gris (transformaciones puntuales)
- tiene menos ruido (filtrado paso bajo)
- algunas estructuras de la imagen aparecen realzadas (filtrado paso alto)

Introducción

El procesamiento de imágenes digitales transforma una imagen en otra que:

- tiene una mejor distribución en los niveles de gris (transformaciones puntuales)
- tiene menos ruido (filtrado paso bajo)
- algunas estructuras de la imagen aparecen realzadas (filtrado paso alto)

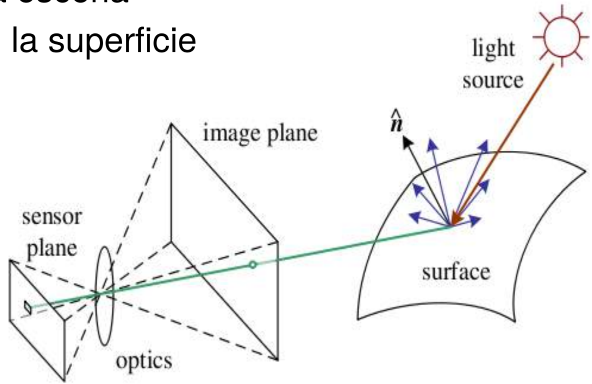


Introducción

- Formación de una imagen digital

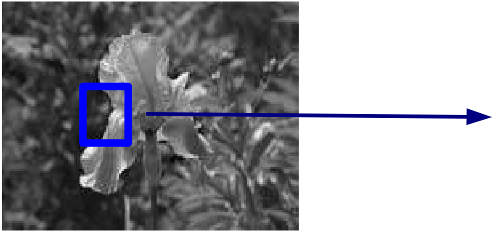
  Los niveles de gris de una imagen dependen:

  - Condiciones de iluminación
  - Geometría de la escena
  - Propiedades de la superficie
  - Cámara

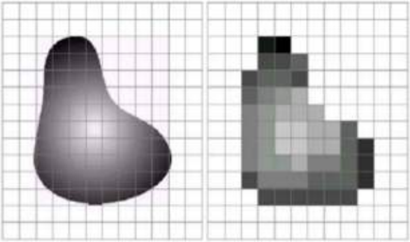


Introducción

- Formación de una imagen digital
  1. Muestrea la retina en una cuadrícula regular
  2. Cuantifica la carga de cada píxel



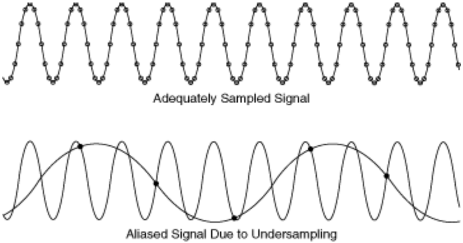
45	60	98	127	132	133	137	133
46	65	98	123	126	128	131	133
47	65	96	115	119	123	135	137
47	63	91	107	113	122	138	134
50	59	80	97	110	123	133	134
49	53	68	83	97	113	128	133
50	50	58	70	84	102	116	126
50	50	52	58	69	86	101	120



Introducción

- Formación de una imagen digital
  1. Muestrea la retina en una cuadrícula regular
  2. Cuantifica la carga de cada píxel

Problema: aliasing

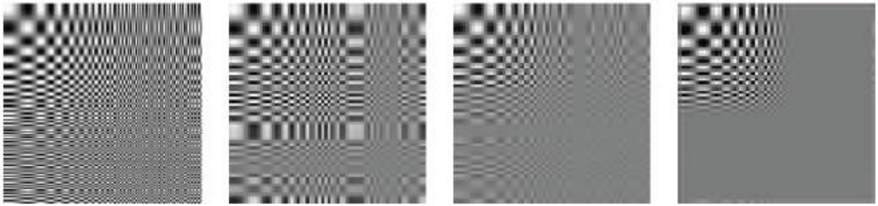


Las componentes de la imagen que no cumplen el teorema de Shannon no se representan adecuadamante.

Introducción

- Formación de una imagen digital
    1. Muestrea la retina en una cuadrícula regular
    2. Cuantifica la carga de cada píxel

Problema: aliasing



Las componentes de la imagen que no cumplen el teorema de Shannon no se representan adecuadamante.

Introducción

- Formación de una imagen digital
  1. Muestrea la retina en una cuadrícula regular
  2. Cuantifica la carga de cada píxel

Problema: aliasing

Introducción

- Imagen digital

Sea $\mathcal{F}$ el conjunto de las filas $\mathcal{C}$ el de las columnas

$$\mathcal{F} = \{0, \ldots, f-1\}$$
$$\mathcal{C} = \{0, \ldots, c-1\}$$

y $f \times c$ la **resolución espacial**.

	0							$c-1$
0	45	60	98	127	132	133	137	133
	46	65	98	123	126	128	131	133
	47	65	96	115	119	123	135	137
	47	63	91	107	113	122	138	134
	50	59	80	97	110	123	133	134
	49	53	68	83	97	113	128	133
	50	50	58	70	84	102	116	126
$f-1$	50	50	52	58	69	86	101	120

Una imagen digital es una función

$$\mathtt{I} : \mathcal{F} \times \mathcal{C} \rightarrow \mathcal{D}$$

siendo

$$\mathcal{D} = \{0, \ldots, d-1\}$$

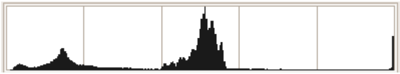
la **resolución digital** de la imagen, habitualmente $d = 256$

Introducción

- Histograma de una imagen digital

  Sea $\mathtt{I}(x, y)$ una imagen digital, su histograma

$$h(m) = \text{card}\{(i,j)|\mathtt{I}(i,j) = m\}, \forall m = 0, \ldots, d-1$$



Introducción

- Histograma de una imagen digital

Sea $\mathtt{I}(x, y)$ una imagen digital, su histograma

$$h(m) = \mathrm{card}\{(i,j) | \mathtt{I}(i,j) = m\}, \forall m = 0, \ldots, d-1$$

Propiedades:

1. $\sum_{m=0}^{d-1} h(m)$ es el número de píxeles
2. $p(k) = \dfrac{h(k)}{\sum_{m=0}^{d-1} h(m)}$ es una fdp
3. El histograma de la unión de un conjunto de regiones disjuntas $R_1 \cup R_2, \cup \ldots \cup R_r$ viene dado por

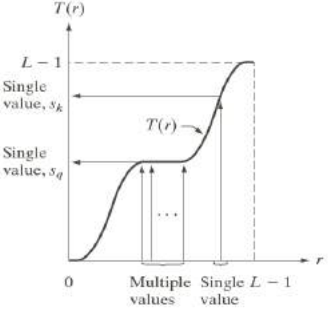
$$h(m) = \sum_i h_{R_i}$$

Transformaciones puntuales

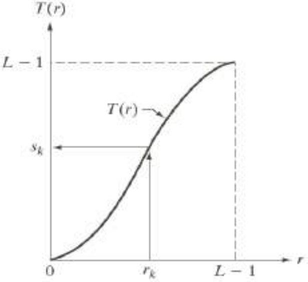
Una transformación puntual $\mathtt{T}$ es una función $\mathtt{T} : \mathcal{D} \to \mathcal{D}$

Toma el nivel de gris de un pixel y lo transforma en otro nivel de gris $\mathtt{T}[\mathtt{I}(x, y)] = \mathtt{S}(x, y) \in \mathcal{D}$

Vamos a asumir que $\mathtt{T}$ es una función monótona no decreciente.

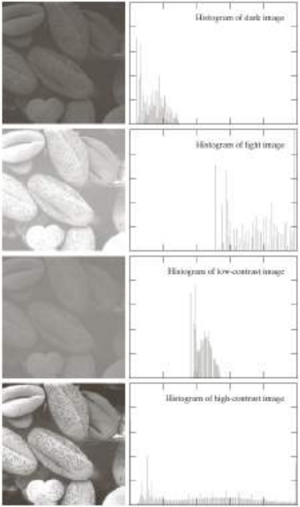


Monótona no decreciente



Monótona creciente

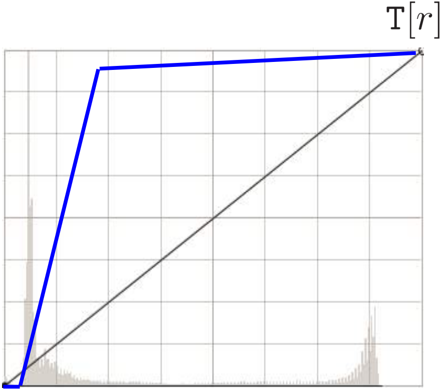
Transformaciones puntuales



Habitualmente deseamos encontrar una transformación que aumente el rango dinámico, y por tanto el contraste, de la imagen.

Transformaciones puntuales

Transformación de una imagen tomada con una iluminación deficiente



Transformaciones puntuales

Transformación de una imagen tomada con una iluminación deficiente

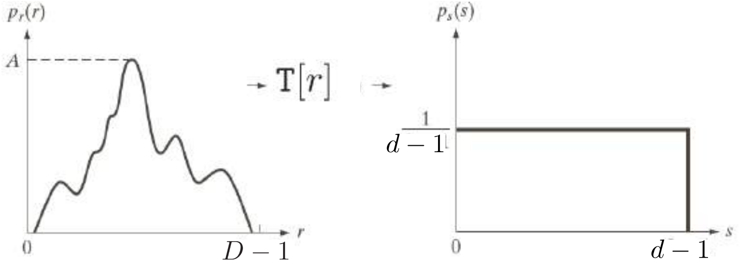
Transformaciones puntuales

Ecualización del histograma

Procedimiento automático de cálculo de `T[r]` de tal forma que en la imagen transformada:

- El rango dinámico sea el máximo
- Los píxeles se distribuyan uniformemente

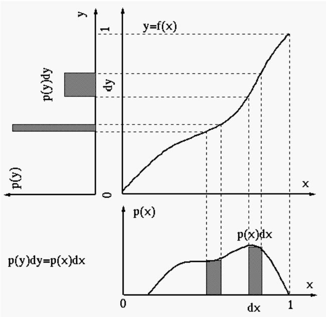
El histograma de la imagen de partida se transformaría en:



Transformaciones puntuales

Ecualización del histograma

1. Establecemos una relación analítica entre el histograma de la imagen de entrada y la de salida.
   Para ello suponemos que $h(k)$ es continua, y $p(s) = \frac{h(k)}{\int_0^1 h(t)dt}$



$$\int_0^r p_i(t)dt = \int_0^{T[r]} p_o(s)ds$$

Transformaciones puntuales

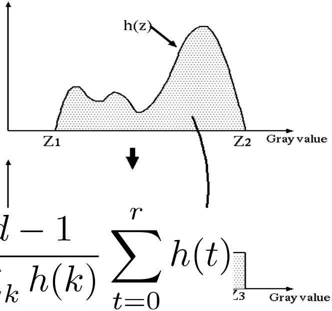
Ecualización del histograma

1. Establecemos una relación analítica entre el histograma de la imagen de entrada y la de salida.
   Para ello suponemos que $h(k)$ es continua, y $p(s) = \frac{h(k)}{\int_0^1 h(t)dt}$

$$\int_0^r p_i(t)dt = \int_0^{T[r]} p_o(s)ds$$

2. Imponemos el histograma de salida.

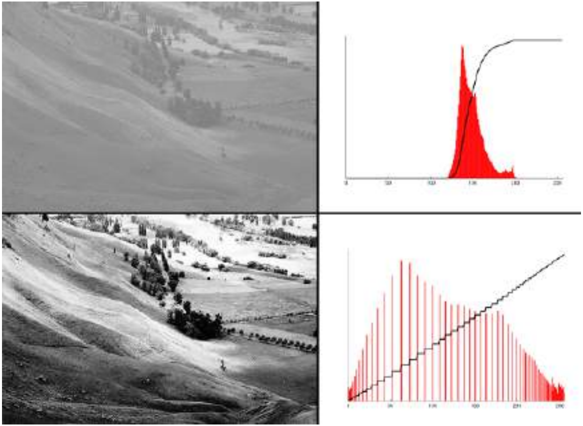
   Para ser uniforme $p_o(x) = \frac{1}{d-1}$



$$T[r] = (d-1)\int_0^r p_i(t)dt = \frac{d-1}{\sum_k h(k)} \sum_{t=0}^{r} h(t)$$

Transformaciones puntuales

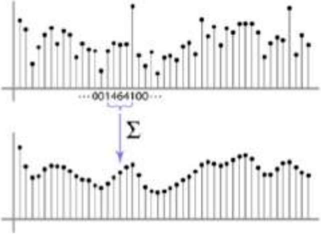
Ecualización del histograma



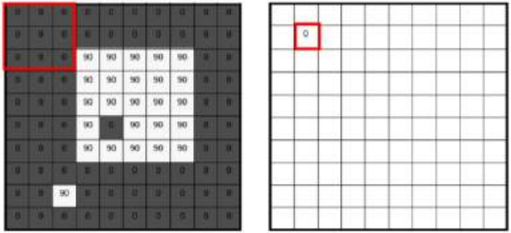
Transformaciones basadas en ventanas

El valor en el que se transforma un píxel depende de su nivel de gris y del de sus vecinos:

- Ventana de vecinos 1-D



- Ventana de vecinos 2-D



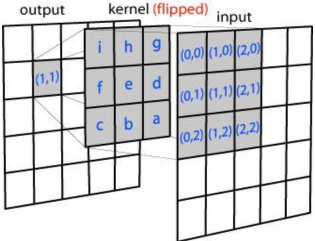
Transformaciones basadas en ventanas

El valor en el que se transforma un píxel depende de su nivel de gris y del de sus vecinos:

- Ventana de vecinos 1-D
- Ventana de vecinos 2-D

Existen dos grupos de transformaciones:

- Transformaciones lineales
  - Los píxeles se transforman mediante una combinación lineal de los valores de los píxeles vecinos.



Transformaciones basadas en ventanas

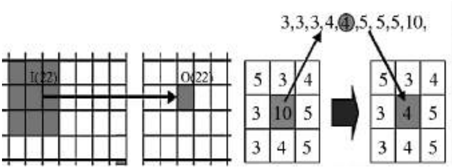
El valor en el que se transforma un píxel depende de su nivel de gris y del de sus vecinos:

- Ventana de vecinos 1-D
- Ventana de vecinos 2-D

Existen dos grupos de transformaciones:

- Transformaciones lineales
- Transformaciones no lineales

    La transformación no puede expresarse como una combinación lineal



Filtrado lineal de imágenes digitales

Convolución

- Unidimensional
  Sean $f(x)$ y $g(y)$ dos funciones discretas definidas en los dominios $x \in \{0, \ldots, a-1\}$, $y \in \{0, \ldots, b-1\}$

$f(x)$	3	4	5		
$g(y)$	1	2	3	4	5

Filtrado de imágenes digitales

Convolución

- Unidimensional
  Sean $f(x)$ y $g(y)$ dos funciones discretas definidas en los dominios $x \in \{0, \ldots, a-1\}$, $y \in \{0, \ldots, b-1\}$.
  La convolución de f con g es otra función $s(r) = f * g$ donde $r \in \{0, \ldots, s-1\}$; $s = a + b - 1$.

$s(r) = f * g$							
$f(x)$	3	4	5				
$g(y)$	1	2	3	4	5		

Filtrado de imágenes digitales

Convolución

- Unidimensional

  Sean $f(x)$ y $g(y)$ dos funciones discretas definidas en los dominios $x \in \{0, \dots, a-1\}, \; y \in \{0, \dots, b-1\}$.

  La convolución de f con g es otra función $s(r) = f * g$ donde $r \in \{0, \dots, s-1\}; \; s = a + b - 1$.

$$s(r) = \sum_{k=0}^{a-1} f(a-1-k) g_e(k + r - (a-1))$$

siendo

$$g_e(y) = \begin{cases} g(y) & \text{si } y \in \{0, \dots, b-1\} \\ 0 & \text{en otro caso} \end{cases}$$

Filtrado de imágenes digitales

Convolución

- Unidimensional

$$s(r) = \sum_{k=0}^{a-1} f(a-1-k)g_e(k+r-(a-1))$$

$$s(r) = f * g \quad \begin{array}{|c|c|c|c|c|c|c|} \hline 3 & & & & & & \\ \hline \end{array}$$

$$r \longrightarrow \quad 0$$

$$f(a-1-k) \quad \begin{array}{|c|c|c|} \hline 5 & 4 & 3 \\ \hline \end{array}$$

$$*$$

$$g_e(k-(a-1)) \quad \begin{array}{|c|c|c|c|c|c|c|} \hline 0 & 0 & 1 & 2 & 3 & 4 & 5 \\ \hline \end{array}$$

Filtrado de imágenes digitales

Convolución

- Unidimensional

$$s(r) = \sum_{k=0}^{a-1} f(a-1-k)\,g_e(k+r-(a-1))$$

$s(r) = f * g$	3	10						
$r \rightarrow$		1						
$f(a-1-k)$			5	4	3			
				$*$				
$g_e(k+1-(a-1))$		0	0	1	2	3	4	5

Filtrado de imágenes digitales

Convolución

- Unidimensional

$$s(r) = \sum_{k=0}^{a-1} f(a-1-k)\,g_e(k+r-(a-1))$$

$$s(r) = f * g \quad \begin{array}{|c|c|c|c|c|c|c|} \hline 3 & 10 & 22 & & & & \\ \hline \end{array}$$

$$r \longrightarrow \qquad 2$$

$$f(a-1-k) \quad \begin{array}{|c|c|c|} \hline 5 & 4 & 3 \\ \hline \end{array}$$

$$*$$

$$g_e(k+2-(a-1)) \quad \begin{array}{|c|c|c|c|c|c|c|} \hline 0 & 0 & 1 & 2 & 3 & 4 & 5 \\ \hline \end{array}$$

Filtrado de imágenes digitales

Convolución

- Unidimensional

$$s(r) = \sum_{k=0}^{a-1} f(a-1-k)g_e(k+r-(a-1))$$

$s(r) = f * g$							
	3	10	22	34			

$r \longrightarrow$ 3

$f(a-1-k)$							
				5	4	3	

$*$

$g_e(k+3-(a-1))$							
	0	0	1	2	3	4	5

Filtrado de imágenes digitales

Convolución

- Unidimensional

$$s(r) = \sum_{k=0}^{a-1} f(a-1-k)g_e(k+r-(a-1))$$

$s(r) = f * g$	3	10	22	34	46			
$r \rightarrow$					4			
$f(a-1-k)$						5	4	3
							$*$	
$g_e(k+4-(a-1))$		0	0	1	2	3	4	5

Filtrado de imágenes digitales

Convolución

- Unidimensional

$$s(r) = \sum_{k=0}^{a-1} f(a-1-k)\,g_e(k+r-(a-1))$$

$s(r) = f * g$

3	10	22	34	46	40	

$r \longrightarrow$ 5

$f(a-1-k)$

5	4	3

$*$

$g_e(k+5-(a-1))$

0	0	1	2	3	4	5	0

Filtrado de imágenes digitales

Convolución

- Unidimensional

$$s(r) = \sum_{k=0}^{a-1} f(a-1-k)\, g_e(k + r - (a-1))$$

$$s(r) = f * g \quad \begin{array}{|c|c|c|c|c|c|c|} \hline 3 & 10 & 22 & 34 & 46 & 40 & 25 \\ \hline \end{array}$$

$r \rightarrow$ 6

$$f(a-1-k) \quad \begin{array}{|c|c|c|} \hline 5 & 4 & 3 \\ \hline \end{array}$$

$*$

$$g_e(k + 6 - (a-1)) \quad \begin{array}{|c|c|c|c|c|c|c|c|c|} \hline 0 & 0 & 1 & 2 & 3 & 4 & 5 & 0 & 0 \\ \hline \end{array}$$

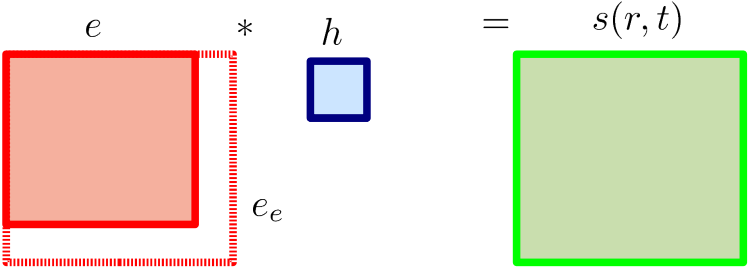
Filtrado de imágenes digitales

Convolución

- Bidimensional

  Sean e y h dos funciones discretas definidas en los dominios $M \times N$ y $m \times n$ respectivamente. La convolución $s = h * e$ es una función de dimensiones $M + m - 1 \times N + n - 1$

$$s(r,t) = \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} h(m-1-i, n-1-j) e_e[r+i-(m-1), t+j-(n-1)]$$



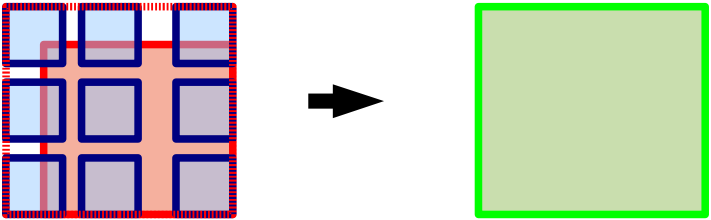
Filtrado de imágenes digitales

Convolución

- Bidimensional

  Sean e y h dos funciones discretas definidas en los dominios $M \times N$ y $m \times n$ respectivamente. La convolución $s = h * e$ es una función de dimensiones $M + m - 1 \times N + n - 1$

$$s(r,t) = \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} h(m-1-i, n-1-j) e_e[r+i-(m-1), t+j-(n-1)]$$



Filtrado de imágenes digitales

Convolución

- Bidimensional

  Si $M \gg m$ y $N \gg n$ es habitual aproximar el resultado de la convolución a

$$s(r,t) = \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} h(m-1-i, n-1-j) e_e[r+i-c_m, t+j-c_n]$$

o esta otra expresión equivalente

$$s(r,t) = \sum_{i=-c_m}^{c_m} \sum_{j=-c_n}^{c_n} h(c_m - i, c_n - j) e[r+i, t+j]$$

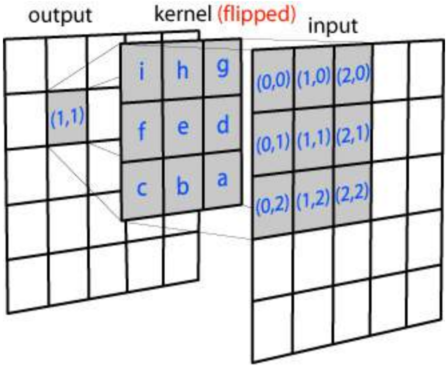
siendo $c_m = \frac{m-1}{2}$ y $c_n = \frac{n-1}{2}$ el centro de masas de h ,

en las que s es de dimensión $M \times N$.

Filtrado de imágenes digitales

Convolución

- Bidimensional
  Si $M \gg m$ y $N \gg n$ es habitual aproximar el resultado de la convolución a



Filtrado de imágenes digitales

Convolución

- Bidimensional

¿Cuál es el resultado de?

Original

$$* \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} =$$

Filtrado de imágenes digitales

Convolución

- Bidimensional

¿Cuál es el resultado de?

Original

$$* \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} =$$

Filtrado de imágenes digitales

Convolución

- Bidimensional

¿Cuál es el resultado de?

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} * \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}$$

Filtrado de imágenes digitales

Convolución

- Propiedades
  - Lineal
    Dadas las funciones g, f , h y los escalares α , β .

$$h * (\alpha f + \beta g) = (\alpha h * f) + (\beta h * g)$$

Por ello al filtrado que se realiza mediante convoluciones también se le conoce como filtrado lineal.

Filtrado de imágenes digitales

Convolución

- Propiedades
  - Lineal
  - Conmutativa
    Dadas las funciones h y g

$$h * g = g * h$$

Esta propiedad no se cumple con la definición simplificada de convolución.

Filtrado de imágenes digitales

Convolución

- Propiedades
  - Lineal
  - Conmutativa
  - Asociativa
    Dadas las funciones f , g y h

$$h * (f * g) = (h * f) * g$$

Filtrado de imágenes digitales

Convolución

- Propiedades
  - Lineal
  - Conmutativa
  - Asociativa
  - Separabilidad

    Dada h una función de dimensión $m \times n$. Decimos que h es separable si existen dos funciones h_x y h_y de dimensiones $m \times 1$ y $1 \times n$ respectivamente, tales que $h = h_x * h_y$.

    La condición necesaria y suficiente para que h sea separable es que $\mathrm{rango}(h) = 1$.

    $h_x * (h_y * f)$ es más eficiente que $(h_x * h_y) * f$

Filtrado de imágenes digitales

Convolución

- Propiedades
  - Lineal
  - Conmutativa
  - Asociativa
  - Separabilidad
  - Vectorización

    Si expresamos la imagen de entrada y de salida en forma vectorial, la convolución puede expresarse como el producto de la imagen por una matriz dispersa.

Filtrado de imágenes digitales

Convolución. Vectorización

Una convolución se puede expresar como un producto matricial.

En una dimensión:

$$\underbrace{[1\ 2]}_{\mathtt{K}_m} * \underbrace{[0\ a\ b\ c\ d\ 0]}_{\mathtt{I}_{M+2p}} = \underbrace{\begin{bmatrix} 1 & 2 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2 & 0 & 0 & 0 \\ 0 & 0 & 1 & 2 & 0 & 0 \\ 0 & 0 & 0 & 1 & 2 & 0 \\ 0 & 0 & 0 & 0 & 1 & 2 \end{bmatrix}}_{\mathcal{T}[\mathtt{K}_m]_{r\times M+2p}} \underbrace{\begin{bmatrix} 0 \\ a \\ b \\ c \\ d \\ 0 \end{bmatrix}}_{\mathtt{I}_{M+2p\times 1}} = \underbrace{\begin{bmatrix} 2a \\ a+2b \\ b+2c \\ c+2d \\ d \end{bmatrix}}_{\mathtt{O}_{r\times 1}}^{\top}$$

donde $\mathcal{T}[\mathtt{K}]$ es la matriz de Toeplitz $\mathtt{K}$ y el tamaño de $\mathtt{O}$

$$r = \frac{M+2p-m}{s} + 1 = \frac{4+2-2}{1} + 1 = 5$$

siendo *p* el “padding” y *s* el “stride” de la convolución (por ahora s=1).

Filtrado de imágenes digitales

Convolución

Una convolución se puede expresar como un producto matricial.

En una dimensión: $\mathtt{K}_m * \mathtt{I}_{M+2p} = \mathcal{T}[\mathtt{K}_m]_{r\times M+2p}\mathtt{I}_{M+2p\times 1} = \mathtt{O}_{r\times 1}$

En dos dimensiones:

$$\underbrace{\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}}_{\mathtt{K}_{m\times n}} * \underbrace{\begin{bmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \end{bmatrix}}_{\mathtt{I}_{M\times N}} \equiv \underbrace{\begin{bmatrix} 1 & 2 & 0 & 0 & 3 & 4 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2 & 0 & 0 & 3 & 4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 2 & 0 & 0 & 3 & 4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 2 & 0 & 0 & 3 & 4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 2 & 0 & 0 & 3 & 4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 0 & 0 & 3 & 4 \end{bmatrix}}_{\mathcal{B}[\mathtt{K}_{m\times n}]_{rt\times MN}} \underbrace{\begin{bmatrix} a \\ b \\ \vdots \\ k \\ l \end{bmatrix}}_{\mathtt{I}_{MN\times 1}} = \underbrace{\begin{bmatrix} a+2b+3e+4f \\ b+2c+3f+4g \\ \vdots \\ f+2g+3j+4k \\ g+2h+3k+4l \end{bmatrix}}_{\mathtt{O}_{rt\times 1}}$$

donde $\mathcal{B}[\mathtt{K}_{m\times n}]_{rt\times MN}$ es la matriz *block-circulant* del kernel $\mathtt{K}_{m\times n}$ y la matriz generada sería be the resultado de cambiar el tamaño de $\mathtt{O}_{rt\times 1}$ to $r \times t$,

siendo
$$r = \frac{M+2p-m}{s} + 1 = \frac{4+0-2}{1} + 1 = 3$$
$$t = \frac{N+2p-n}{s} + 1 = \frac{3+0-2}{1} + 1 = 2$$

Filtrado de imágenes digitales

Filtrado lineal

- Definición
  Aquella transformación que puede expresarse como la convolución de la imagen con un filtro.

Vamos a estudiar dos grupos de filtros:

- **Suavizado (filtrado paso bajo)**
  Atenúan el ruido que contamina la imagen.
  Se basan en promediar los niveles de gris.

- **Extracción de bordes (filtrado paso alto)**
  Realzan las transiciones de niveles de gris.
  Se basan en calcular diferencias entre los niveles de gris.

Filtrado de imágenes digitales

Filtrado lineal

Suavizado de imágenes digitales

- Filtro de promediado “tipo caja”

  El valor en el que se transforma un píxel es el promedio de sus vecinos.

Por ejemplo:

$$h_c^{3\times 3} = \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$h_c^{3\times 3} = h_c^{1\times 3} * h_c^{3\times 1} = \frac{1}{3}\begin{bmatrix} 1 & 1 & 1 \end{bmatrix} * \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}\frac{1}{3}$$

Filtrado de imágenes digitales

Filtrado lineal

Suavizado de imágenes digitales

- Filtro de promediado “tipo caja”

  El valor en el que se transforma un píxel es el promedio de sus vecinos.

En general:

$$h_c^{n\times n} = h_c^{1\times n} * h_c^{n\times 1}$$

donde

$$h_c^{1\times n} = \frac{1}{n}\begin{bmatrix} 1 & \cdots & 1 \end{bmatrix} \text{ y } h_c^{n\times 1} = \frac{1}{n}\begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}$$

Filtrado de imágenes digitales

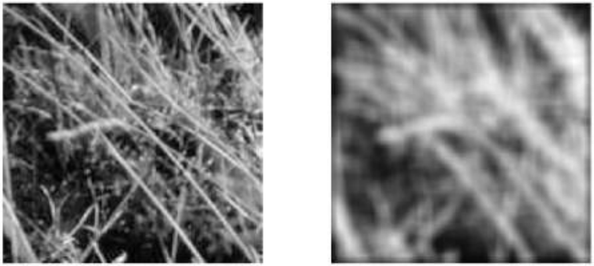
Filtrado lineal

Suavizado de imágenes digitales

- Filtro de promediado “tipo caja”

  El valor en el que se transforma un píxel es el promedio de sus vecinos.

  Aunque suaviza una imagen, su comportamiento no es perfecto, pues los píxeles cercanos y lejanos influyen de igual manera en el resultado.



Filtrado de imágenes digitales

Filtrado lineal

Suavizado de imágenes digitales

- Filtro de suavizado gaussiano

  Realiza un promediado ponderado por la función gaussiana

$$g(x) = e^{-\frac{1}{2}\left[\frac{x}{\sigma}\right]^2}$$

  donde x es la distancia al píxel filtrado y σ es la desviación típica del filtro.

  En este caso, el filtro unidimensional sería

$$h_g^n(x) = e^{-\frac{1}{2}\left[\frac{x-\frac{n-1}{2}}{\sigma}\right]^2} \cdot \left[\sum_{k=0}^{n-1} e^{-\frac{1}{2}\left(\frac{k-\frac{n-1}{2}}{\sigma}\right)^2}\right]^{-1}$$

  El bidimensional $h_g^{n\times n} = h_{gx}^n * h_{gy}^n$

  Existe una relación entre n y σ: $n \geq 5\sigma$

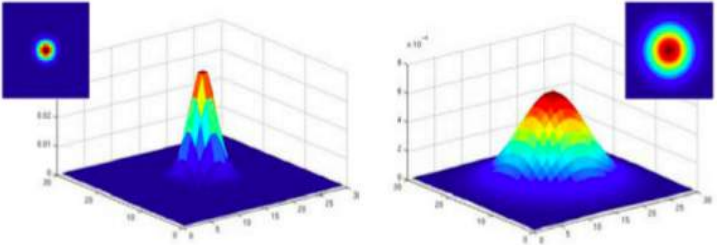
Filtrado de imágenes digitales

Filtrado lineal

Suavizado de imágenes digitales

- Filtro de suavizado gaussiano

  La varianza del filtro determina la fuerza del suavizado



$\sigma = 2$ with 30 x 30 kernel

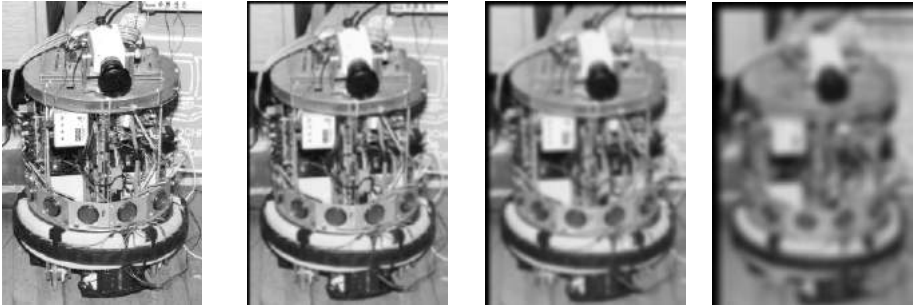
$\sigma = 5$ with 30 x 30 kernel

Filtrado de imágenes digitales

Filtrado lineal

Suavizado de imágenes digitales

- Filtro de suavizado gaussiano



$\sigma = 0$ $\sigma = 1$ $\sigma = 2$ $\sigma = 4$

Filtrado de imágenes digitales

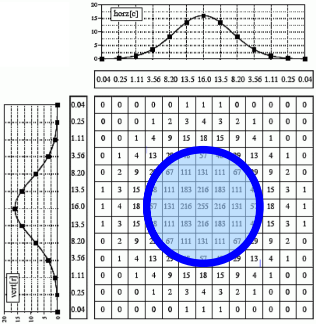
Filtrado lineal

Suavizado de imágenes digitales

- Filtro de suavizado gaussiano

La forma del filtro definido es

$$h_g^{n \times n} =$$



Circular o **isotrópica** (filtra con igual fuerza en todas las direcciones). No sólo elimina ruido, sino también suaviza los bordes.

Filtrado de imágenes digitales

Filtrado lineal

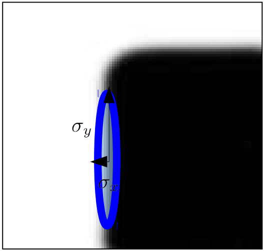
Suavizado de imágenes digitales

- Filtro de suavizado gaussiano

  El filtro gaussiano **anisotrópico** filtra con distinta fuerza en cada dirección.

$$h_{g_a}^{m\times n} = h_{g_a x}^{1\times n} * h_{g_a y}^{m\times 1}$$

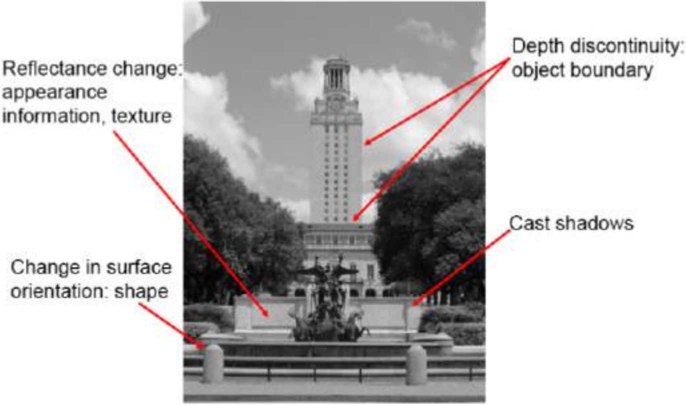
$$n = 5\sigma_x$$
$$m = 5\sigma_y$$



Filtrado de imágenes digitales

Filtrado lineal

Extracción de bordes ¿Por qué? ¿Qué causa un borde?



Filtrado de imágenes digitales

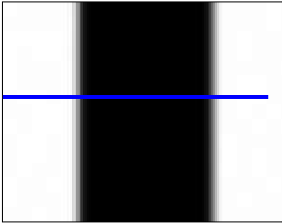
Filtrado lineal

Extracción de bordes

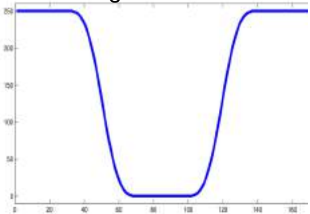
- Filtro basado en el gradiente

  Los bordes de una imagen están en los máximos del gradiente.

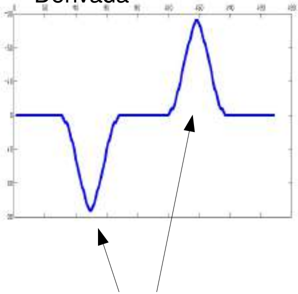
Imagen



Perfil de grises



Derivada



Los bordes coinciden con los extremos de la derivada

Filtrado de imágenes digitales

Filtrado lineal

Extracción de bordes

- Filtro basado en el gradiente

  El gradiente de la imagen $\mathrm{I}(x, y)$ es un vector

$$\bar{\nabla}\mathrm{I} = \begin{bmatrix} \frac{\partial \mathrm{I}}{\partial x} \\ \\ \frac{\partial \mathrm{I}}{\partial y} \end{bmatrix} \equiv \begin{bmatrix} \mathrm{I}_x \\ \mathrm{I}_y \end{bmatrix}$$

  - El módulo informa sobre la existencia de borde

$$|\bar{\nabla}\mathrm{I}| = \sqrt{\mathrm{I}_x^2 + \mathrm{I}_y^2}$$

  - La dirección informa sobre la orientación del borde

$$\theta = \arctan\left(\frac{\mathrm{I}_y}{\mathrm{I}_x}\right)$$

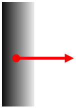
Filtrado de imágenes digitales

Filtrado lineal

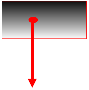
Extracción de bordes

- Filtro basado en el gradiente

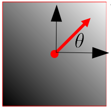
  El gradiente de la imagen $\mathrm{I}(x, y)$ es un vector que apunta en la dirección de máximo crecimiento de la intensidad



$$\nabla \mathbf{I} = \left[\frac{\partial \mathrm{I}}{\partial x}, 0\right]$$



$$\nabla \mathbf{I} = \left[0, \frac{\partial \mathrm{I}}{\partial y}\right]$$



$$\nabla \mathbf{I} = \left[\frac{\partial \mathrm{I}}{\partial x}, \frac{\partial \mathrm{I}}{\partial y}\right]$$

$$|\bar{\nabla}\mathrm{I}| = \sqrt{\mathrm{I}_x^2 + \mathrm{I}_y^2} \qquad \theta = \arctan\left(\frac{\mathrm{I}_y}{\mathrm{I}_x}\right)$$

Filtrado de imágenes digitales

Filtrado lineal

Extracción de bordes

- Filtro basado en el gradiente

¿Cómo calculo el gradiente de una imagen?

$$\frac{\partial \mathtt{I}(x,y)}{\partial x} \equiv \mathtt{I}_x \approx \frac{\mathtt{I}(x+\delta x, y) - \mathtt{I}(x-\delta x, y)}{2\delta x}$$

$$\mathtt{I}_x \approx \frac{1}{2}\left[\mathtt{I}(x+1,y) - \mathtt{I}(x-1,y)\right] = h_{dx} * \mathtt{I}$$

$$h_{dx} = \frac{1}{2}\left[\begin{array}{ccc} 1 & 0 & -1 \end{array}\right] \quad h_{dy} = \frac{1}{2}\left[\begin{array}{c} 1 \\ 0 \\ -1 \end{array}\right]$$

luego $\mathtt{I}_x = h_{dx} * \mathtt{I}$ e $\mathtt{I}_y = h_{dy} * \mathtt{I}$

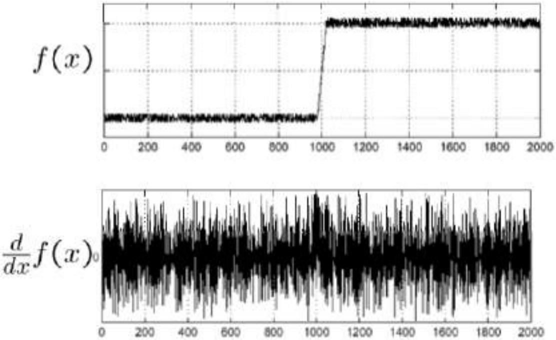
Filtrado de imágenes digitales

Filtrado lineal

Extracción de bordes

- Filtro basado en el gradiente

  ¿Cómo afecta el ruido a la derivada?



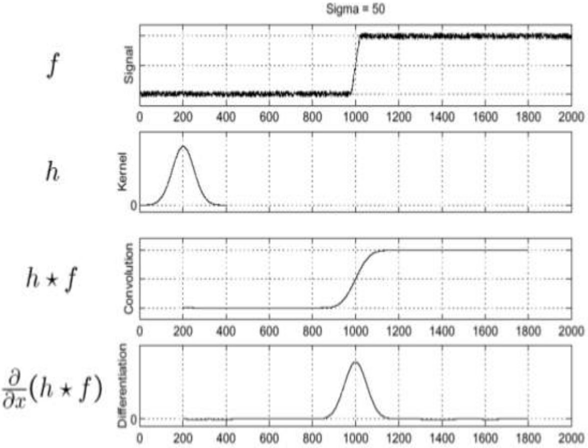
Filtrado de imágenes digitales

Filtrado lineal

Extracción de bordes

- Filtro basado en el gradiente

  ¿Cómo afecta el ruido a la derivada?



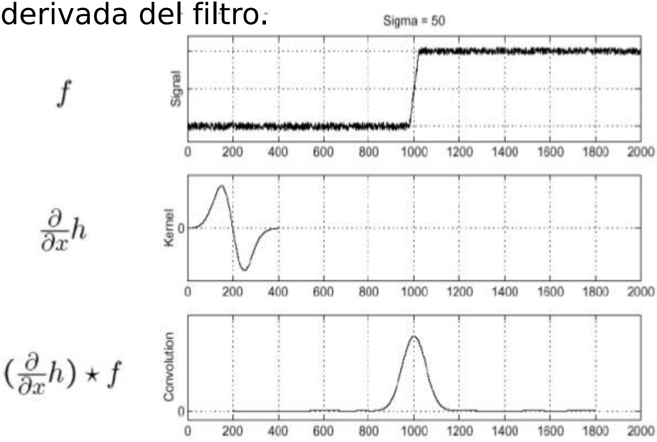
Filtrado de imágenes digitales

Filtrado lineal

Extracción de bordes

- Filtro basado en el gradiente

  La derivada de una convolución es igual que la convolución con la derivada del filtro.



Filtrado de imágenes digitales

Filtrado lineal

Extracción de bordes

- Filtro basado en el gradiente

  En el caso bidimensional se puede mejorar el comportamiento promediando en la dirección ortogonal a la que derivamos.

  - Promediado tipo "caja" (filtro de Prewitt)

$$\mathrm{I}_x = h_{dx} * h_c^{3\times1} * \mathrm{I} = \frac{1}{2}[1 \quad 0 - 1] * \frac{1}{3}\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} * \mathrm{I} = \frac{1}{6}\begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} * \mathrm{I}$$

  - Promediado binomial (filtro de Sobel)

$$\mathrm{I}_x = h_{dx} * h_s^{3\times1} * \mathrm{I} = \frac{1}{2}[1 \quad 0 - 1] * \frac{1}{4}\begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} * \mathrm{I} = \frac{1}{8}\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} * \mathrm{I}$$

Filtrado de imágenes digitales

Filtrado lineal

Extracción de bordes

- Filtro basado en el gradiente

  En el caso bidimensional se puede mejorar el comportamiento promediando en la dirección ortogonal a la que derivamos.

  - Derivada del gaussiano bidimensional

$$g'(x) = -\frac{x}{\sigma^2} e^{-\frac{1}{2}\left[\frac{x}{\sigma}\right]^2} \Rightarrow h_{dg}^n(x) = -\left[\frac{x - \frac{n-1}{2}}{\sigma^2}\right] e^{-\frac{1}{2}\left[\frac{x - \frac{n-1}{2}}{\sigma}\right]^2}$$

Por tanto, las componentes horizontal y vertical del gradiente para una imagen I

$$\mathrm{I}_x = h_{dgx}^n * h_{gy}^n * \mathrm{I} = \mathcal{D}_x * \mathrm{I}$$

$$\mathrm{I}_y = h_{dgy}^n * h_{gx}^n * \mathrm{I} = \mathcal{D}_y * \mathrm{I}$$

Filtrado de imágenes digitales

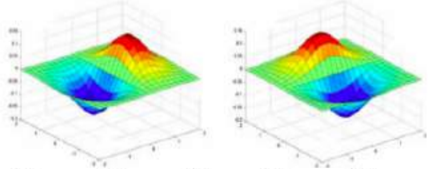
Filtrado lineal

Extracción de bordes

- Filtro basado en el gradiente

  En el caso bidimensional se puede mejorar el comportamiento promediando en la dirección ortogonal a la que derivamos.

  - <u>Derivada del gaussiano bidimensional</u>



$\mathcal{D}_x = h^n_{dgx} * h^n_{gy}$ $\quad$ $\mathcal{D}_y = h^n_{dgy} * h^n_{gx}$

Filtrado de imágenes digitales

Filtrado lineal

Extracción de bordes

- Filtro basado en las derivadas direccionales

  Calculan la derivada de la imagen en una dirección $\mathbf{u}$

  $$\mathbf{u} \cdot \nabla \mathrm{I}(\mathbf{x})$$

  siendo $\mathbf{u} = (\cos\theta, \sin\theta)$

  En la práctica se pueden calcular rotando un ángulo θ las matrices empleadas en el cálculo del gradiente

  $$[\mathbf{u}(\theta) \cdot \nabla] * \mathrm{I}(\mathbf{x})$$

  La derivada del gaussiano tiene la propiedad de "orientabilidad"

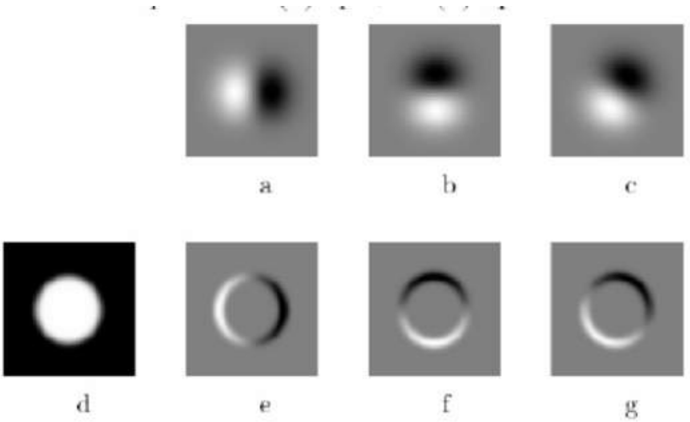
  $$[\mathbf{u}(\theta) \cdot \nabla] * \mathrm{I}(\mathbf{x}) = \cos\theta \mathcal{D}_x * \mathrm{I}(\mathbf{x}) + \sin\theta \mathcal{D}_y * \mathrm{I}(\mathbf{x})$$

Filtrado de imágenes digitales

Filtrado lineal

Extracción de bordes

- Filtro basado en las derivadas direccionales



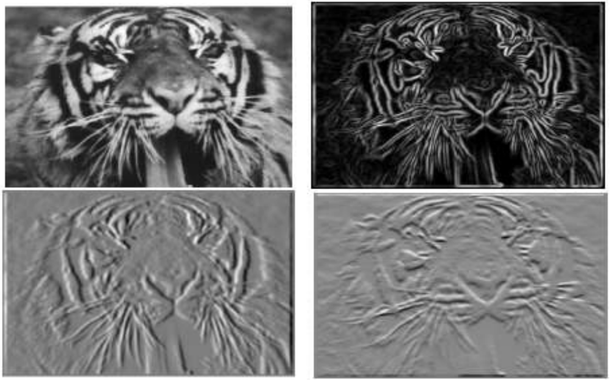
Filtrado de imágenes digitales

Filtrado lineal

Extracción de bordes

- Filtro basado en el gradiente

$\mathtt{I}$ $\quad\quad$ $|\nabla \mathtt{I}|$



$\mathtt{I}_x = \mathbf{h}_{dx} * \mathtt{I}$ $\quad\quad$ $\mathtt{I}_y = \mathbf{h}_{dy} * \mathtt{I}$

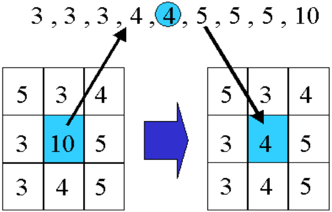
Filtrado no lineal

- Filtrado de la mediana

  El valor transformado es la mediana de los vecinos.

  Filtra bien los valores atípicos, pero no aquellos contaminados con ruido gaussiano.

  Adecuado para filtrar el ruido impulsivo.

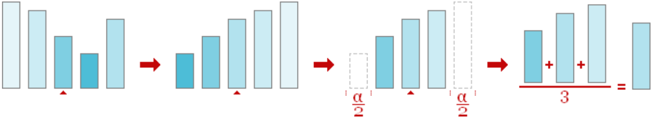


added noise

median

Filtrado no lineal

- **Filtrado de la media truncada** (trimmed mean)
  El valor transformado es la media de los vecinos, eliminando la fracción más alejada.



- **Filtrado bilateral** (transf. lineal dependiente del píxel)
  En vez de eliminar una fracción fija de valores, el valor resultante depende de su distancia en la imagen y su diferencia de nivel de gris al píxel central.

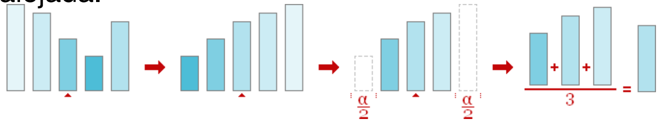
$$I^{\text{filtered}}(x) = \frac{1}{W_p} \sum_{x_i \in \Omega} I(x_i) f_r(\|I(x_i) - I(x)\|) g_s(\|x_i - x\|),$$

donde $\{f_r(x), g_s(x)\} = \mathcal{N}(x|\{\sigma_r, \sigma_s\})$

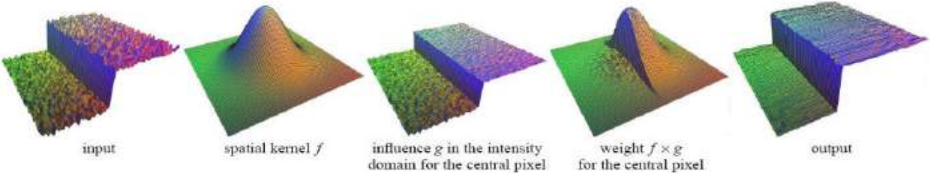
$$W_p = \sum_{x_i \in \Omega} f_r(\|I(x_i) - I(x)\|) g_s(\|x_i - x\|)$$

Filtrado no lineal

- **Filtrado de la media truncada** (trimmed mean)
  El valor transformado es la media de los vecinos, eliminando la fracción más alejada.

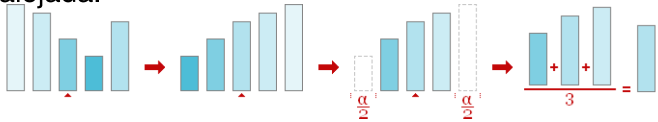


- **Filtrado bilateral** (transf. lineal dependiente del píxel)
  En vez de eliminar una fracción fija de valores, el valor resultante depende de su distancia en la imagen y su diferencia de nivel de gris al píxel central.



Filtrado no lineal

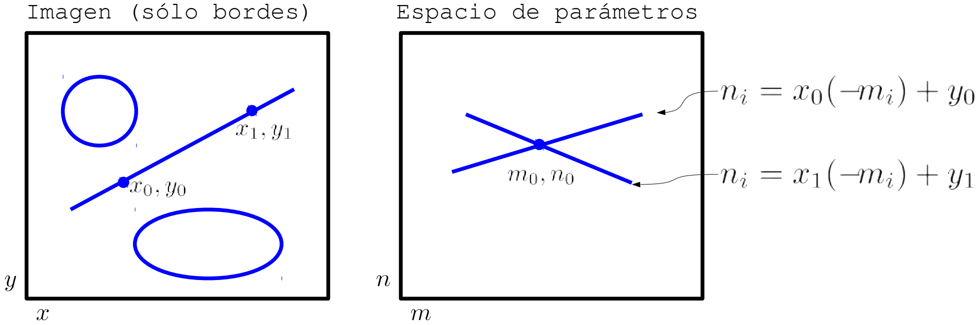
- **Filtrado de la media truncada** (trimmed mean)
  El valor transformado es la media de los vecinos, eliminando la fracción más alejada.



- **Filtrado bilateral** (transf. lineal dependiente del píxel)
  En vez de eliminar una fracción fija de valores, el valor resultante depende de su distancia en la imagen y su diferencia de nivel de gris

La transformada Hough

Localiza una curva en una imagen detectando máximos locales en el espacio de parámetros de la curva.



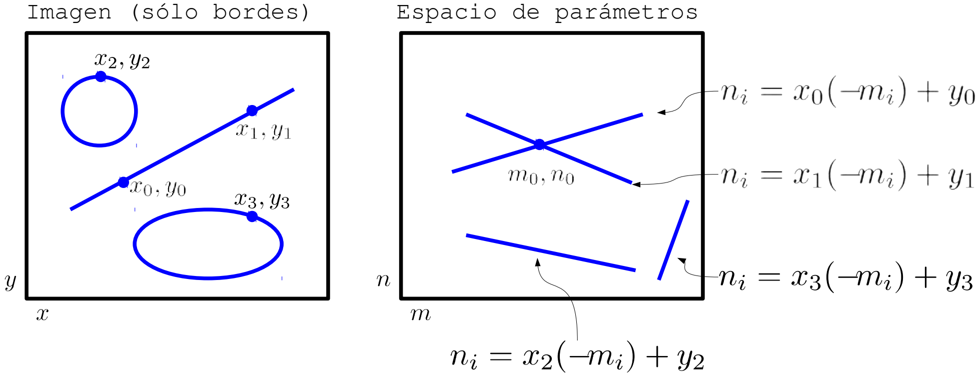
$$y_i = m_0 x_i + n_0 \longrightarrow (m_0, n_0)$$

$$(x_0, y_0) \longleftarrow n_i = x_0(-m_i) + y_0$$

Cada punto de la imagen define una curva en el espacio de parámetros.

La transformada Hough

Localiza una curva en una imagen detectando máximos locales en el espacio de parámetros de la curva.



Cada punto de la imagen define una curva en el espacio de parámetros.

La transformada Hough

Localiza una curva en una imagen detectando máximos locales en el espacio de parámetros de la curva.

Algoritmo:

1.Discretizo el espacio de parámetros

$$m \in \{-m_0, \ldots, +m_0\};\ n \in \{-n_0, \ldots, +n_0\}$$

2.Inicio $P(m,n) = 0,\ \forall m, n$

3.Para todo píxel $\mathrm{I}(x,y) \neq 0$

Para todo $m = -m_0, \ldots, +m_0$

$$n = x(-m) + y$$

$$P(m,n) + +$$

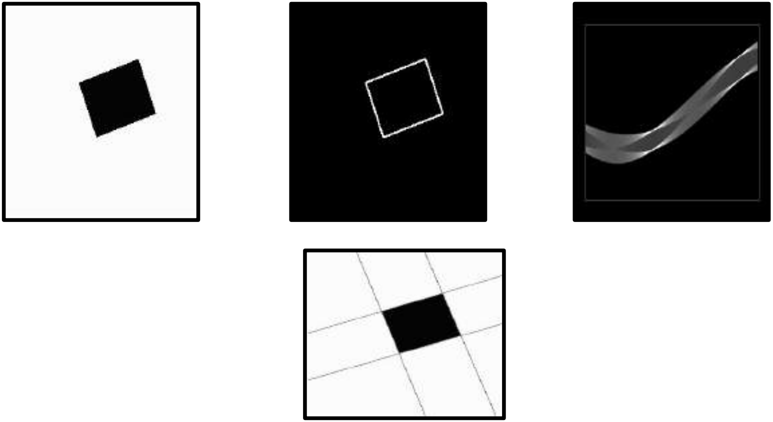
4.Busca los máximos locales de $P(m,n)$

Problema: m y n pueden hacerse ∞

Solución: $\rho = x\cos\theta + y\sin\theta$

La transformada Hough

Localiza una curva en una imagen detectando máximos locales en el espacio de parámetros de la curva.
Ejemplo:



La transformada Hough

Localiza una curva en una imagen detectando máximos locales en el espacio de parámetros de la curva.

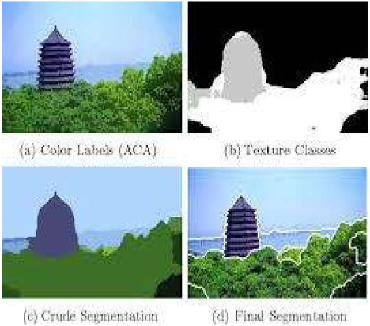
Discusión:

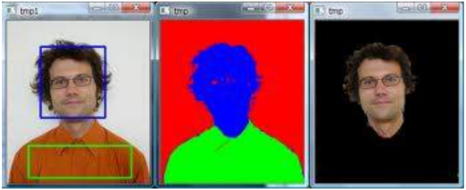
- Encuentra cualquier curva paramétrica $y = f(x, \theta)$
- Detecta todas las instancias en una pasada
- Robusto al ruido
- Los recursos computacionales necesarios crecen exponencialmente con el número de parámetros.

Segmentación de imágenes digitales

Concepto:

- Divide una imagen en segmentos o regiones que se corresponden con estructuras de la imagen.
- Cada píxel tiene asociada una etiqueta.
- Es una representación más abstracta de la escena, paso intermedio reconstrucciones, reconocimiento (chicken-egg).

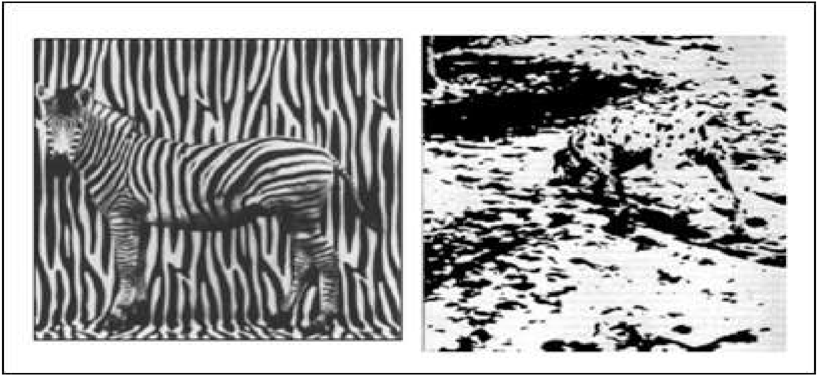




Segmentación de imágenes digitales

Concepto:

- Divide una imagen en segmentos o regiones que se corres-ponden con estructuras de la imagen.
- Cada píxel tiene asociada una etiqueta.
- Es una representación más abstracta de la escena, paso intermedio reconstrucciones, reconocimiento (chicken-egg).



Segmentación de imágenes digitales

Concepto:

- Divide una imagen en segmentos o regiones que se corresponden con estructuras de la imagen.
- Cada píxel tiene asociada una etiqueta.
- Es una representación más abstracta de la escena, paso intermedio reconstrucciones, reconocimiento (chicken-egg).

- Sólo consideraremos segmentación bottom-up. Ingnoramos la contribución del reconocimiento o la estructura.
- Sólo utilizaremos la apariencia (valores de los píxeles), aunque también puede utilizarse otros datos como movimiento, disparidad (estructura), etc.

Segmentación de imágenes digitales

Aproximación:

- Segmentar ≡ Etiquetar ≡ Clasificar
- Describimos los píxeles de la imagen con un vector de características discriminantes:

$$\mathbf{F}(\mathbf{x}) = (\mathbf{x}, I(\mathbf{x}), \mathbf{L}(\mathbf{x}))^\top$$

Tipos:

- Supervisada

  Existen ejemplos de imágenes segmentadas para entrenar el clasificador
- No supervisada

  Busca agrupaciones naturales de los píxeles.

Leer Zseliski, Cap. 5

Segmentación de imágenes digitales

Métodos basados en grafos:

- Plantean el problema a partir de la minimización de una energía:

$$E = \sum_{i=1}^{n} D_i(x_i) + \lambda \sum_{ij\ldots m \in \mathcal{V}} V_{ij\ldots m}(x_i, x_j, \ldots, x_m)$$

donde:

$D_i(x_i)$ coste de clasificar píxel *i* en clase x_i

$V_{ij\ldots m}$ coste de que los píxeles *i,j...m* tengan etiquetas x_i,x_j,...

λ factor de escala entre ambos términos.

x_i son las posibles etiquetas del píxel *i*-ésimo.

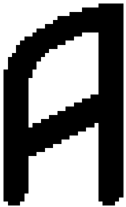
En algunos casos el mínimo de esta energía se puede obtener como el flujo que satura a una red.

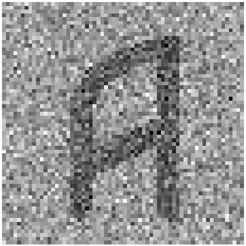
Segmentación de imágenes digitales

Métodos basados en cortes de grafos

1. Planteamiento del problema

- Una imagen binaria sólo tiene dos posible niveles de gris. Deseamos eliminar el ruido que contamina dicha imagen.



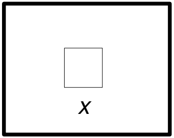


Segmentación de imágenes digitales

Métodos basados en cortes de grafos

1. Planteamiento del problema

- Una imagen binaria sólo tiene dos posible niveles de gris. Deseamos eliminar el ruido que contamina dicha imagen. Supongamos que la imagen sólo tiene 1 píxel, *x*



$$x = \begin{cases} 0 & \text{objeto} \\ 1 & \text{fondo} \end{cases}$$

- Planteo el problema de segmentación como una optimización.

Segmentación $\equiv$ Optimización energía, $E(x)$

- Conozco $D(x)$: coste incurrido al asignar a x una etiqueta

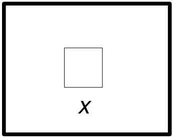
$$D(x) = \begin{cases} \text{- valor alto si etiqueta incorrecta} \\ \text{- valor bajo si etiqueta correcta} \end{cases}$$

Segmentación de imágenes digitales

Métodos basados en cortes de grafos

1. Planteamiento del problema

- Una imagen binaria sólo tiene dos posible niveles de gris. Deseamos eliminar el ruido que contamina dicha imagen. Supongamos que la imagen sólo tiene 1 píxel, *x*



$$x = \begin{cases} 0 & \text{objeto} \\ 1 & \text{fondo} \end{cases}$$

- Planteo el problema de segmentación como una optimización.

  Segmentación $\equiv$ Optimización energía, $E(x)$

- Conozco $D(x)$: coste incurrido al asignar a x una etiqueta

$$D(x) = \begin{cases} D(0) = 2 \\ D(1) = 5 \end{cases}$$

Segmentación de imágenes digitales

Métodos basados en cortes de grafos

1. Planteamiento del problema

- Si la función de energía es $E(x) = D(x)$, la etiqueta óptima para el píxel x será

$$\hat{x} = \operatorname*{argmin}_y D(y) \quad \Rightarrow \quad \hat{x} = 0$$

- Si la imagen tiene más píxeles $\mathbf{x} = (x_1, x_2, \ldots, x_n)$

$$E(\mathbf{x}) = \sum_{i=1}^{n} D_i(x_i)$$

- Puedo definir los costes en esta imagen

$$D(x) = \begin{cases} D_i(0) = \text{nivel de gris de } x_i \equiv I(x_i) \\ D_i(1) = 255 - \text{nivel de gris de } x_i \equiv 255 - I(x_i) \end{cases}$$

- El mínimo

$$\hat{\mathbf{x}} = \operatorname*{argmin}_{\mathbf{y}} E(\mathbf{y}) = (\operatorname*{argmin}_{y_1} D_1(y_1), \ldots, \operatorname*{argmin}_{y_n} D_n(y_n))$$

Segmentación de imágenes digitales

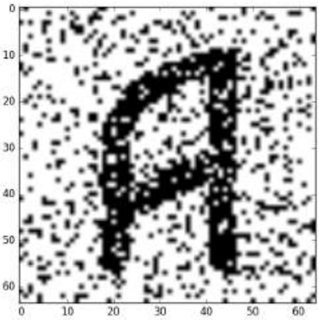
Métodos basados en cortes de grafos

1. Planteamiento del problema

- El mínimo

$$\hat{\mathbf{x}} = \underset{\mathbf{y}}{\operatorname{argmin}}\, E(\mathbf{y}) = (\underset{y_1}{\operatorname{argmin}}\, D_1(y_1), \ldots, \underset{y_n}{\operatorname{argmin}}\, D_n(y_n))$$

$$\hat{\mathbf{x}} = \underset{\mathbf{y}}{\operatorname{argmin}}\, E(\mathbf{y}) = \begin{cases} 1 \text{ si } I(y_i) > 128 \\ 0 \text{ si } I(y_i) < 128 \end{cases}$$



Segmentación de imágenes digitales

Métodos basados en cortes de grafos

1. Planteamiento del problema

- El mínimo

$$\hat{\mathbf{x}} = \operatorname*{argmin}_{\mathbf{y}} E(\mathbf{y}) = (\operatorname*{argmin}_{y_1} D_1(y_1), \ldots, \operatorname*{argmin}_{y_n} D_n(y_n))$$

- **Problema:**

Estamos suponiendo que x_i y x_j son independientes, incluso cuando son vecinos $x_i \in \mathcal{N}(x_j)$

- **Solución:**

Introducir un nuevo término de energía que tenga en cuenta las etiquetas de dos píxeles vecinos

Segmentación de imágenes digitales

Métodos basados en cortes de grafos

1. Planteamiento del problema

- Solución:

$$E(\mathbf{x}) = \sum_{i=1}^{n} D_i(x_i) + \lambda \sum_{ij \in \mathcal{V}} V_{ij}(x_i, x_j)$$

donde

$D_i(x_i)$ coste de clasificar píxel *i* en clase x_i

$V_{ij}(x_i, x_j)$ coste de que los píxeles *i,j* tengan etiquetas x_i, x_j

λ factor de escala entre ambos términos.

$x_i \in \{0, 1\}$ son las posibles etiquetas del píxel *i*-ésimo.

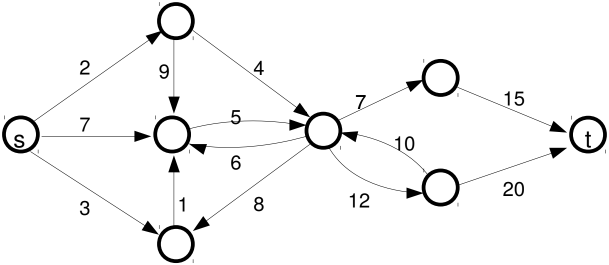
- ¿Cómo se minimiza?
  - Hasta finales de los ‘90 se creía que era un problema NP-duro.
  - Algunas $E(\mathbf{x})$ pueden resolverse en tiempo polinómico empleando una equivalencia con la maximización de flujo en grafos.

Segmentación de imágenes digitales

Métodos basados en cortes de grafos

2. Optimización de flujo en redes

- <u>Red de flujo.</u>
  Grafo dirigido $\mathbb{G}(\mathcal{N}, \mathcal{E})$, siendo $\mathcal{N}, \mathcal{E}$ conjuntos de **nodos** y **arcos**, con dos nodos especiales, $s \in \mathcal{N}$ **fuente** y $t \in \mathcal{N}$ **sumidero**.



- Cada arco tiene asociado un **coste** $\mathcal{C} : \mathcal{E} \to \mathbb{R}^+$ que representa la cantidad de flujo que admite.

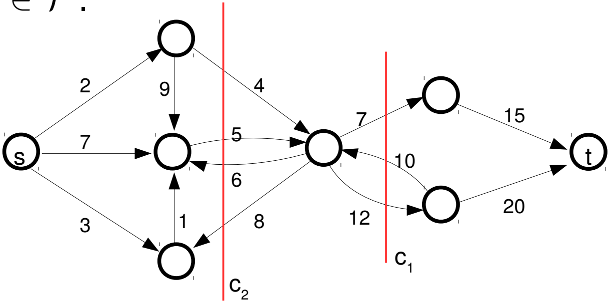
Segmentación de imágenes digitales

Métodos basados en cortes de grafos

2. Optimización de flujo en redes

- Corte s-t.

  Partición de $\mathcal{N}$ en dos conjuntos disjuntos $\mathcal{S}, \mathcal{T}$, tales que $s \in \mathcal{S}$ y $t \in \mathcal{T}$.



- Conjunto de corte.

  Arcos que “cruzan” el corte, con origen en $\mathcal{S}$ y destino en $\mathcal{T}$.

$$\{(u,v) \in \mathcal{E} : u \in \mathcal{S}, v \in \mathcal{T}\}$$

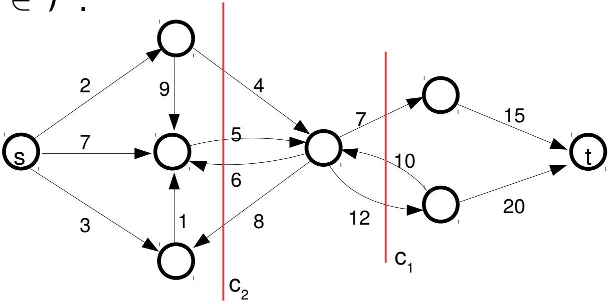
Segmentación de imágenes digitales

Métodos basados en cortes de grafos

2. Optimización de flujo en redes

- <u>Corte s-t.</u>

  Partición de $\mathcal{N}$ en dos conjuntos disjuntos $\mathcal{S}, \mathcal{T}$, tales que $s \in \mathcal{S}$ y $t \in \mathcal{T}$.



- <u>Coste de un corte.</u>

  Suma de las capacidades de los arcos en el conjunto de corte.

  coste(c_1) = 19;   coste (c_2) = 9

Segmentación de imágenes digitales

Métodos basados en cortes de grafos

2. Optimización de flujo en redes

- <u>Teorema de flujo máximo corte mínimo (max-flow min-cut).</u>
  El máximo flujo a través de la red, desde *s* a *t*, viene dado por el coste del corte mínimo.

$$\boxed{\text{Flujo máximo} \equiv \text{coste corte mínimo}}$$

- Existen algoritmos eficientes de cálculo del flujo máximo
  - Método de Ford-Fulkerson
  - Shortest augmenting path
  - Push-relabel ...

- Algunos problemas de optimización pueden modelarse de tal forma que el coste de cada corte se corresponda con una asignación de valor a las variables del problema.

Segmentación de imágenes digitales

Métodos basados en cortes de grafos

3. Minimización de energía mediante cortes de grafos

- Existe un grafo $\mathbb{G}$ que modela el coste de $E(\mathbf{x})$.

$$E(\mathbf{x}) = \sum_{i=1}^{n} D_i(x_i) + \lambda \sum_{ij \in \mathcal{V}} V_{ij}(x_i, x_j)$$

si se cumple que:

- $x_i \in \mathcal{L} = \{0, 1\}$, el problema sólo tiene dos etiquetas.
- $V_{ij} : \{0, 1\}^2 \rightarrow \mathbb{R}^+$, sólo existen restricciones binarias.
- V_{ij} es **submodular**

$$V_{ij}(0,0) + V_{ij}(1,1) \leq V_{ij}(0,1) + V_{ij}(1,0)$$

Segmentación de imágenes digitales

Métodos basados en cortes de grafos

3. Minimización de energía mediante cortes de grafos

- Existe un grafo $\mathbb{G}$ que modela el coste de $E(\mathbf{x})$.

  Construcción del grafo:

  - Incluyo tantos nodos como píxeles tiene la imagen

    $$\mathcal{N} = \{s, t\} \cup \{x_1, \ldots, x_n\}$$

  - Por cada nodo x_i incluyo una arista $s \rightarrow x_i$ con coste $D_i(0)$.
  - Por cada nodo x_i incluyo una arista $t \leftarrow x_i$ con coste $D_i(1)$.
  - Por cada pareja $ij \in \mathcal{V}$ incluyo las aristas
    - $x_i \rightarrow x_j$ con coste $V_{ij}(1,0)$
    - $x_i \leftarrow x_j$ con coste $V_{ij}(0,1)$
  - Interpretamos
    - $x_i \in \mathcal{S}$ entonces $x_i = 1$
    - $x_i \in \mathcal{T}$ entonces $x_i = 0$

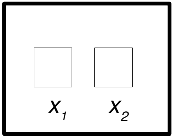
Segmentación de imágenes digitales

Métodos basados en cortes de grafos

3. Minimización de energía mediante cortes de grafos

- <u>Ejemplo.</u>

  Sea la imagen anterior, pero con dos píxeles: $x_i \in \{0, 1\}$



$$D_1(0) = 2 \quad D_2(0) = 9 \qquad V_{12}(1,0) = 1; V_{12}(0,1) = 2$$
$$D_1(1) = 5 \quad D_2(1) = 4 \qquad V_{12}(0,0) = V_{12}(1,1) = 0$$