

Transformadas de distancias discretas: Algoritmos y aplicaciones

Antoine Manzanera
ENSTA
ParisTech

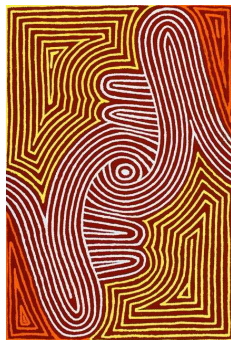
Objetivos del curso

- Introducción a la geometría discreta.
 - Distancia discreta
 - Topología discreta
- Transformada de distancia discreta: Algoritmos
 - Distancias relacionadas a la topología
 - Distancias casi euclidianas
- Transformada de distancia discreta: Aplicaciones
 - Análisis de imágenes: filtrado, segmentación
 - Representación de patrones: código, reconocimiento

Problemática

La función distancia o «Transformada de distancia» es relacionada a un objeto (o conjunto, o patrón) X en un espacio E . Asocia a cada elemento de X su distancia al complementario de X en E : $X^c = E \setminus X$.

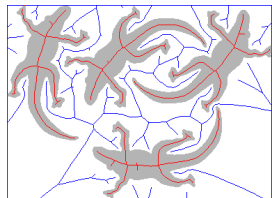
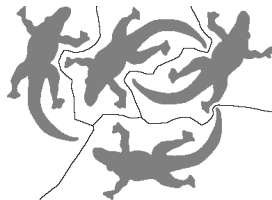
Esta función está obviamente vinculada a una distancia d definida sobre $E \times E$. Nos interesamos en el cálculo de esta función en el espacio E discreto, así como su uso.



Narjula Scobie Narrapula - *Women's
body ceremony* (Australia)

Aplicaciones

- Procesamiento de imágenes.
 - Filtrados: operadores morfológicos
 - Segmentación: Zonas de influencia
 - Evolución de regiones o contornos activos
- Representación de patrones.
 - Esqueletos
 - Código, compresión
 - Reconocimiento, indexación



Índice

- 1 Formalismo, Definiciones y Notaciones
 - Teselaciones y Imágenes discretas
 - Topologías en la malla cuadrada
 - Distancias en \mathbb{Z}^n
- 2 Transformadas de distancia: Algoritmos
 - Algoritmos básicos
 - Distancias casi euclidianas
- 3 Aplicaciones de las transformadas de distancia
 - Operadores morfológicos
 - Esqueleto morfológico y erosión última
 - Esqueletos conexos multi escala
- 4 Conclusión

Índice

- 1 Formalismo, Definiciones y Notaciones
 - Teselaciones y Imágenes discretas
 - Topologías en la malla cuadrada
 - Distancias en \mathbb{Z}^n
- 2 Transformadas de distancia: Algoritmos
 - Algoritmos básicos
 - Distancias casi euclidianas
- 3 Aplicaciones de las transformadas de distancia
 - Operadores morfológicos
 - Esqueleto morfológico y erosión última
 - Esqueletos conexos multi escala
- 4 Conclusión

Teselación del espacio

Una teselación es una *partición* del espacio \mathbb{R}^n en un conjunto de partidas $\{P_i\}_{i \in I}$:

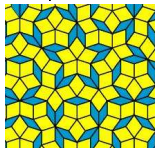
- $\bigcup_{i \in I} P_i = \mathbb{R}^n$
- $\forall (i, j) \in I^2, P_i \cap P_j = \emptyset$

Las P_i son denominadas *teselas* (o píxeles, o vóxeles).

Pajarita - Alhambra (Granada)



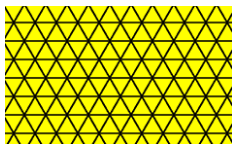
Teselación aperiódica de Penrose



Teselación del plano

En el caso del plano \mathbb{R}^2 , solo existen 3 *teselaciones regulares*, es decir que respetan las condiciones siguientes:

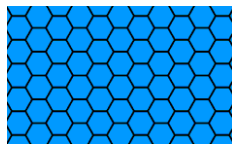
- los P_i son todos idénticos.
- los P_i son polígonos regulares (i.e. convexos, lados iguales, ángulos iguales).
- cada vértice de P_i está en contacto con otros vértices.



Teselación triangular



Teselación cuadrada

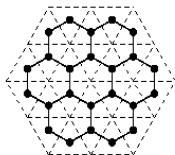


Teselación hexagonal

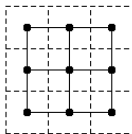
Teselación y grafo

A toda teselación se puede asociar un *grafo* cuyos *vértices* representan las teselas, y cuyas *aristas* representan la relación de adyacencia entre teselas (2 teselas son *adyacentes* si comparten un lado). Un tal grafo es denominado *mall* del plano.

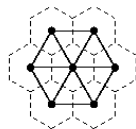
Teselación y mallas regulares son representaciones duales del plano discreto:



Malla hexagonal
(Tesela triangular)



Malla cuadrada
(Tesela cuadrada)



Malla triangular
(Tesela hexagonal)

Propiedades de las teselaciones regulares

La elección del tipo de teselación regular en análisis de imágenes se hace según ciertas propiedades:

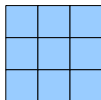
- Conformidad a la geometría del sensor.
- Recursividad (multi-resolución).
- Número de direcciones representadas.
- Extensión a las dimensiones superiores.
- Representación en \mathbb{Z}^n .

La malla cúbica

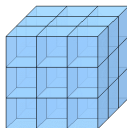
Por esas razones, la malla cuadrada (en 2d), o cúbica (en 3d) es la más usada en análisis de imágenes.

El *espacio discreto* está, pues, representado por \mathbb{Z}^n .

Un *píxel* (2d) o *vóxel* (3d) es un elemento de \mathbb{Z}^n .



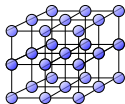
Teselación 2d



Teselación 3d



Malla 2d



Malla 3d

Imágen binaria

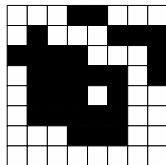
Sea \mathbb{Z}^n el espacio discreto.

Una imágen binaria I está definida como un subconjunto de \mathbb{Z}^n :

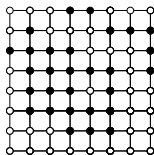
$$I \subset \mathbb{Z}^n$$

El píxel $p \in \mathbb{Z}^n$ está representado en negro si y solo si $p \in I$.

Presentación «Teselas»



Presentación «Malla»



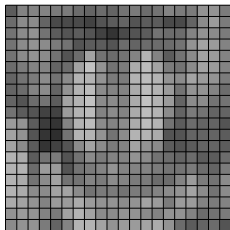
Imágen digital

Sea \mathbb{Z}^n el espacio discreto.

Una imágen digital F está definida como una función de \mathbb{Z}^n de valores enteras:

$$F : \mathbb{Z}^n \rightarrow \mathbb{N}$$

El píxel $p \in \mathbb{Z}^n$ está representado con un nivel de gris proporcional a $F(p)$.



Topologías en la malla cuadrada

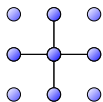
La topología en las imágenes discretas está definida a partir de la relación de conectividad inducida por el grafo de la malla (X, S) , donde X representa los vértices y S las aristas.

$$X \subset \mathbb{Z}^2; S \subset X^2$$

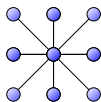
Sean x e y 2 puntos de X , por definición x e y son *adyacentes* si:

$$x \approx y \Leftrightarrow (x, y) \in S$$

En la malla cuadrada, 2 tipos de relaciones de adyacencia pueden ser considerados:



4-connectividad

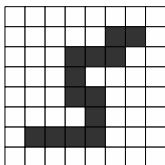


8-connectividad

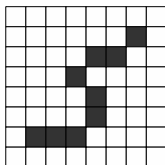
Caminos y componentes conexos

La cerradura transitiva de la relación de adyacencia \approx es una relación de equivalencia: « existe un *camino conexo* entre x e y » :

$$x \sim y \Leftrightarrow \exists \{x_1, \dots, x_n\}, x \approx x_1, \dots, x_i \approx x_{i+1}, \dots, x_n \approx y$$



Camino 4-conexo

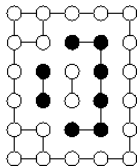
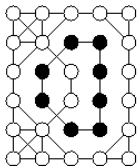


Camino 8-conexo

Las clases de equivalencia de la relación « \sim » se denominan las *componentes conexas* de X .

Huecos y Teorema de Jordan

En la malla cuadrada, la noción de hueco en un conjunto X ($X \subset \mathbb{Z}^2$), que debe corresponder a una componente conexa finita del complementario X^c , no está bien definida...



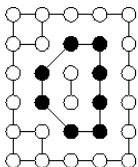
8-conectividad

4-conectividad

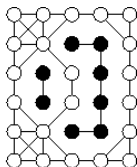
Este problema está vinculado a la validez del teorema de Jordan, según el cual una curva simple cerrada separa el plano en 2 componentes conexas, una de las cuales sea finita.

Huecos y Teorema de Jordan

...a no ser que se considere diferentes tipos de conectividad para X y para X^c :



8-conectividad



4-conectividad

El Teorema de Jordan es válido por la (8-4)-conectividad y por la (4,8)-conectividad.

Huecos y Teorema de Jordan

Ejemplo : *¿ Cuantas componentes conexas, y cuantos huecos, (1) por la (8-4)-conectividad, (2) por la (4,8)-conectividad ?*



Distancias y bolas discretas

Distancia en \mathbb{Z}^n

$d : \mathbb{Z}^n \times \mathbb{Z}^n$ es una *distancia*

$$\Leftrightarrow \begin{cases} d(x, y) = d(y, x) \\ d(x, y) = 0 \Leftrightarrow x = y \\ d(x, z) \leq d(x, y) + d(y, z) \end{cases}$$

Bola discreta

Sea d una distancia de \mathbb{Z}^n , $x \in \mathbb{Z}^n$, $r \in \mathbb{N}$.

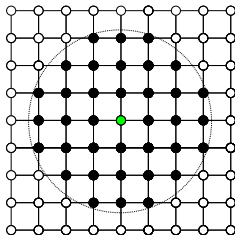
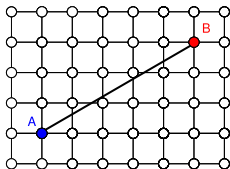
La *bola* de centro x y de radio r está definida por:

$$B_r^d(x) = \{y \in \mathbb{Z}^n; d(x, y) \leq r\}.$$

Distancia euclidiana

Distancia euclidiana en \mathbb{Z}^2

$$d_E(x, y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2}$$



$$d_E(A, B) = \sqrt{5^2 + 3^2} = \sqrt{34} \quad B_{\sqrt{10}}^{d_E}(C) = \{y \in \mathbb{Z}^n; d_E(C, y) \leq \sqrt{10}\}$$

Distancia d_4

Si se define la distancia entre 2 puntos x e y como la largura del camino conexo mas corto entre x e y , la topología 4-conexa induce una distancia d_4 . Ponderando todas las aristas del grafo de la malla por el valor 1, resulta:

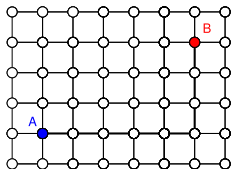
Distancia d_4 en \mathbb{Z}^2

$$d_4(x, y) = |x_1 - y_1| + |x_2 - y_2|$$

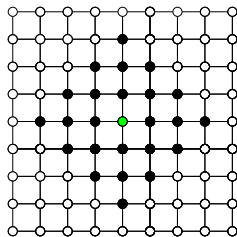
Distancia d_4

Distancia d_4 en \mathbb{Z}^2

$$d_4(x, y) = |x_1 - y_1| + |x_2 - y_2|$$



$$d_4(A, B) = 5 + 3 = 8$$



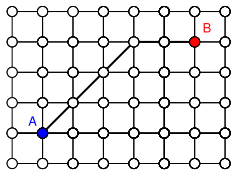
$$B_3^{d_4}(C) = \{y \in \mathbb{Z}^n; d_4(C, y) \leq 3\}$$

Distance d_8

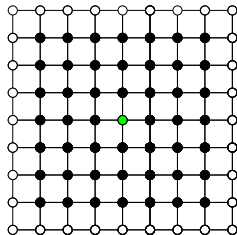
Del mismo modo la topología 8-conexa induce una distancia d_8 definida por:

Distancia d_8 en \mathbb{Z}^2

$$d_8(x, y) = \max(|x_1 - y_1|, |x_2 - y_2|)$$



$$d_4(A, B) = \max(5, 3) = 5$$



$$B_3^{d_8}(C) = \{y \in \mathbb{Z}^n; d_8(C, y) \leq 3\}$$

Índice

- 1 Formalismo, Definiciones y Notaciones
 - Teselaciones y Imágenes discretas
 - Topologías en la malla cuadrada
 - Distancias en \mathbb{Z}^n
- 2 Transformadas de distancia: Algoritmos
 - Algoritmos básicos
 - Distancias casi euclidianas
- 3 Aplicaciones de las transformadas de distancia
 - Operadores morfológicos
 - Esqueleto morfológico y erosión última
 - Esqueletos conexos multi escala
- 4 Conclusión

Transformada de distancia

Transformada de distancia

Sea d una distancia en \mathbb{Z}^n , $X \subset \mathbb{Z}^n$ una imagen binaria.

La *transformada de distancia* f_d^X está definida por :

$$\begin{aligned} f_d^X : \mathbb{Z}^n &\rightarrow \mathbb{N} \\ x &\mapsto d(x, X^c) \end{aligned}$$

Transformada de distancia d_4

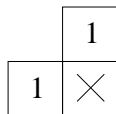
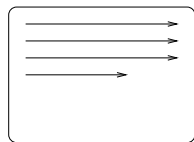
Sea X una imagen binaria de dimensiones $W \times H$. La transformada de distancia d_4 sobre X está calculada por el algoritmo recursivo siguiente, según 2 secuencias sucesivas :

(1) Secuencia causal

```

for j = 1 to H
  for i = 1 to W
    if  $(i, j) \in X$  :  $F_4^X(i, j) = \infty$ ;
    else  $F_4^X(i, j) = 0$ 
       $F_4^X(i, j) = \min( F_4^X(i, j),$ 
                      $F_4^X(i - 1, j) + 1,$ 
                      $F_4^X(i, j - 1) + 1);$ 
    endfor
  endfor

```



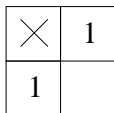
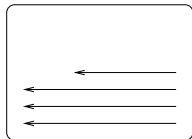
Transformada de distancia d_4

(2) Secuencia anticausal

```

for j = H downto 1
  for i = W downto 1
     $F_4^X(i, j) = \min( F_4^X(i, j),$ 
                       $F_4^X(i + 1, j) + 1,$ 
                       $F_4^X(i, j + 1) + 1);$ 
  endfor
endfor

```



La complejidad del algoritmo es de 2 comparaciones por píxel.

Transformada de distancia d_4 : ilustración

0	0	0	0	0	0	0	0	0
0	∞	∞	∞	∞	∞	∞	∞	0
0	∞	∞	∞	∞	∞	∞	∞	0
0	∞	∞	0	0	∞	∞	∞	0
0	∞	∞	∞	∞	∞	∞	∞	0
0	∞	∞	∞	∞	∞	∞	∞	0
0	∞	∞	∞	∞	∞	∞	∞	0
0	0	0	0	0	0	0	0	0

(0) Inicialización

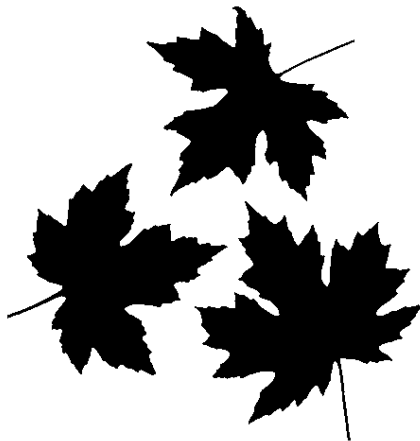
0	0	0	0	0	0	0	0	0
0	1	1	1	1	1	1	1	0
0	1	2	2	2	2	2	2	0
0	1	2	0	0	1	2	0	0
0	1	2	1	1	2	3	0	0
0	1	2	2	2	3	4	0	0
0	0	0	0	0	0	0	0	0

(1) Después secuencia causal

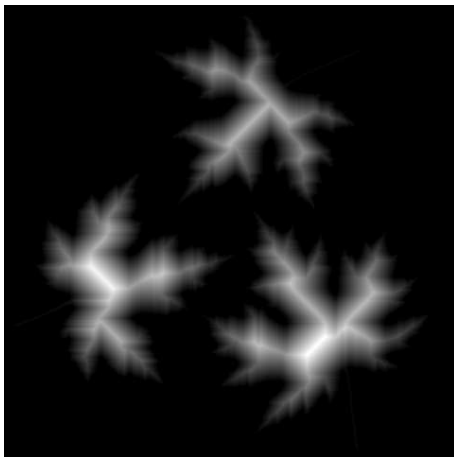
0	0	0	0	0	0	0	0	0
0	1	1	1	1	1	1	1	0
0	1	2	1	1	2	1	0	0
0	1	1	0	0	1	1	0	0
0	1	2	1	1	2	1	0	0
0	1	1	1	1	1	1	0	0
0	0	0	0	0	0	0	0	0

(2) Después 2 secuencias

Transformada de distancia d_4 : ilustración



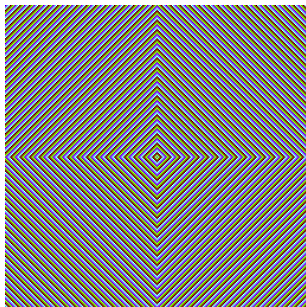
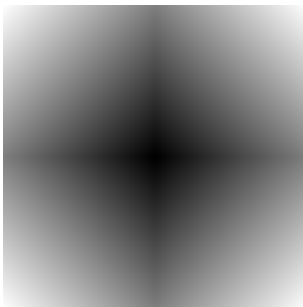
Transformada de distancia d_4 : ilustración



Transformada de distancia d_4 : ilustración

Izquierda: transformada de distancia d_4 del complementario de un conjunto reducido a un píxel (centro de la imagen).

Derecha: líneas de niveles r (en color) de la transformada de distancia : círculos discretos de radios r .



Transformada de distancia d_8

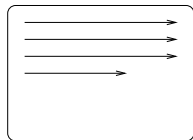
Del mismo modo, la transformada de distancia d_8 sobre una imagen binaria X se calcula de la manera siguiente:

(1) Secuencia causal

```

for j = 1 to H
  for i = 1 to W
    if  $(i, j) \in X : F_8^X(i, j) = \infty;$ 
    else  $F_8^X(i, j) = 0;$ 
     $F_8^X(i, j) = \min( F_8^X(i, j),$ 
       $F_8^X(i - 1, j - 1) + 1,$ 
       $F_8^X(i, j - 1) + 1,$ 
       $F_8^X(i + 1, j - 1) + 1,$ 
       $F_8^X(i - 1, j) + 1);$ 
  endfor
endfor

```



1	1	1
1	×	

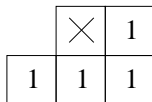
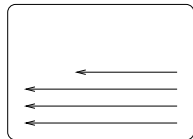
Transformada de distancia d_8

(2) Secuencia anticausal

```

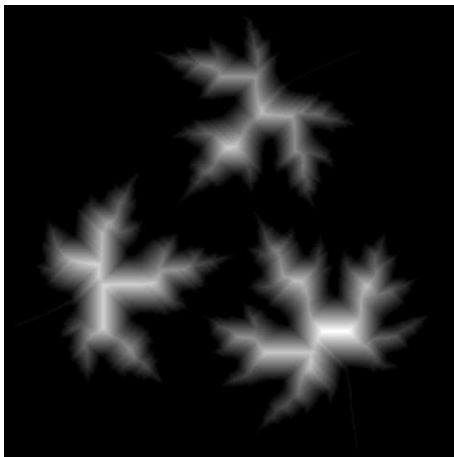
for j = H downto 1
  for i = W downto 1
     $F_8^X(i, j) = \min( F_8^X(i, j),$ 
       $F_8^X(i + 1, j + 1) + 1,$ 
       $F_8^X(i, j + 1) + 1,$ 
       $F_8^X(i - 1, j + 1) + 1,$ 
       $F_8^X(i + 1, j) + 1);$ 
  endfor
endfor

```



La complejidad del algoritmo es de 4 comparaciones por píxel.

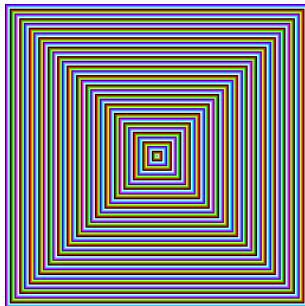
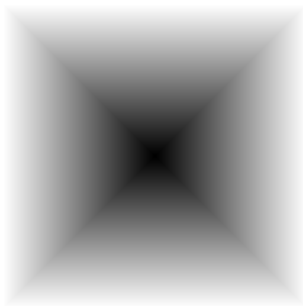
Transformada de distancia d_g : ilustración



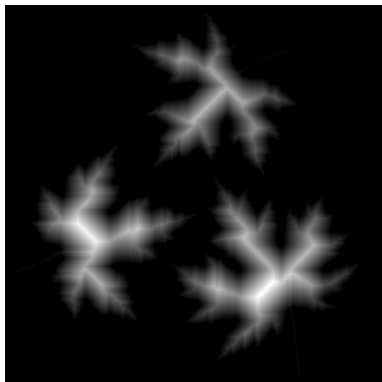
Transformada de distancia d_8 : ilustración

Izquierda: transformada de distancia d_8 del complementario de un conjunto reducido a un píxel (centro de la imagen).

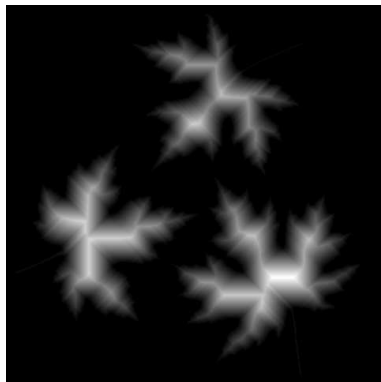
Derecha: líneas de niveles r (en color) de la transformada de distancia : círculos discretos de radios r .



Comparación distancias d_4 y d_8



transformada de distancia d_4

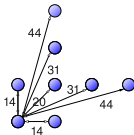
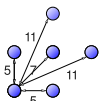


transformada de distancia d_8

Distancias de chamfer

Las distancias de chamfer son una tentativa por reducir el carácter anisotrópico de las distancias d_4 et d_8 , y por acercarse de la distancia euclidiana. El principio consiste también en definir la distancia como la largura del camino más corto entre dos puntos, pero con una ponderación de las aristas diferente según el tipo. Por ejemplo, la distancia de chamfer d_{3-4} (arriba)

También se puede considerar mallas más complejas, i.e. vecindades más grandes: d_{5-7-11} (medio), o aún $d_{14-20-31-44}$ (abajo).



Transformada de distancia d_{3-4}

Por ejemplo, la transformada de distancia d_{3-4} sobre una imagen binaria X se calcula de la manera siguiente:

(1) Secuencia causal

for $j = 1$ to H

 for $i = 1$ to W

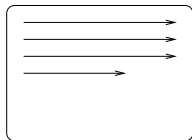
 if $(i, j) \in X : F_{3-4}^X(i, j) = \infty;$

 else $F_{3-4}^X(i, j) = 0;$

$F_{3-4}^X(i, j) = \min(F_{3-4}^X(i, j),$
 $F_{3-4}^X(i - 1, j - 1) + 4,$
 $F_{3-4}^X(i, j - 1) + 3,$
 $F_{3-4}^X(i + 1, j - 1) + 4,$
 $F_{3-4}^X(i - 1, j) + 3);$

 endfor

endfor



4	3	4
3	×	

Transformada de distancia d_{3-4}

(2) Secuencia anticausal

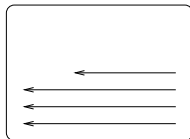
for $j = H$ downto 1

 for $i = W$ downto 1

$$F_{3-4}^X(i, j) = \frac{1}{3} \times \min(F_{3-4}^X(i, j), \\ F_{3-4}^X(i+1, j+1) + 4, \\ F_{3-4}^X(i, j+1) + 3, \\ F_{3-4}^X(i-1, j+1) + 4, \\ F_{3-4}^X(i+1, j) + 3);$$

 endfor

endfor

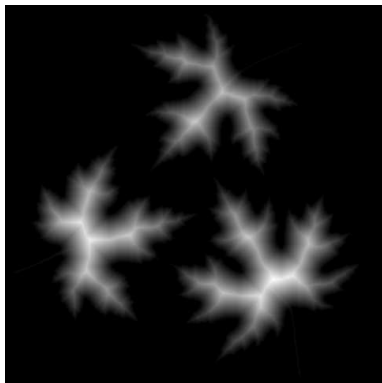


	×	3
4	3	4

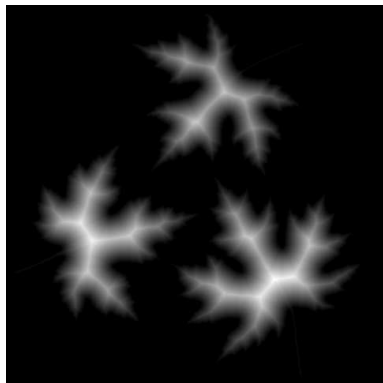
La complejidad del algoritmo es de 4 comparaciones por píxel.

Observe la división por 3 para normalizar.

Distancias de chamfer: ilustraciones



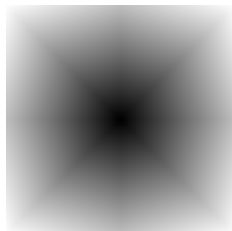
transformada de distancia d_{3-4}



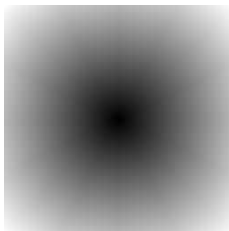
transformada de distancia d_{5-7-11}

Distancias de chamfer: ilustraciones

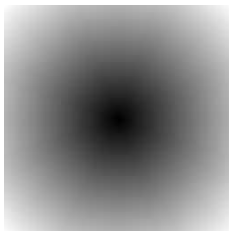
4	3	4
3	×	



	11		11	
11	7	5	7	11
	5	×		

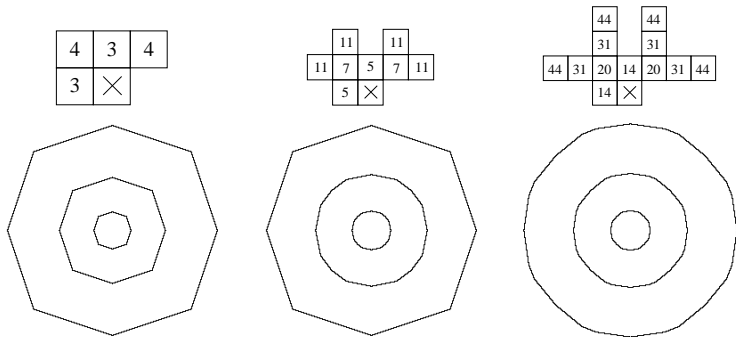


		44		44		
		31		31		
44	31	20	14	20	31	44
		14	×			



Máscaras de cálculo (causal) y transformada de distancia del complementario del píxel central, por las 3 primeras distancias de chamfer.

Distancias de chamfer: ilustraciones

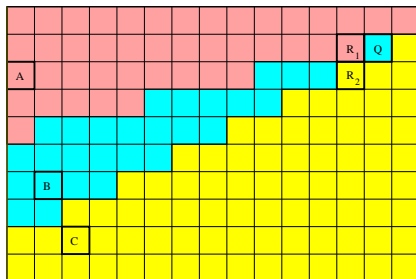


Máscaras de cálculo (causal) et círculos discretos (radios 25, 75 et 125), por las 3 primeras distancias de chamfer.

Transformada de distancia euclidiana

Problema: ¿ Puede calcularse una transformada de distancia euclidiana *exacta* sobre cualquiera imagen binaria X por una secuencia de pasadas involucrando únicamente cálculos en la vecindad 3×3 ?

Respuesta : ¡ NO !



Transformada de distancia euclidiana

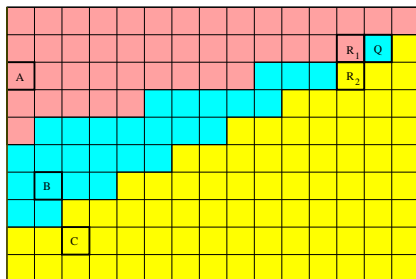
$$d_E(R_1, A) < d_E(R_1, B),$$

$$d_E(R_2, C) < d_E(R_2, B) \text{ pero}$$

$$d_E(Q, B) < d_E(Q, A) \text{ y}$$

$$d_E(Q, B) < d_E(Q, C).$$

Entonces no se puede decidir localmente de la transformada de distancia en el punto Q .



Algoritmo de Danielsson-Leymarie

Sin embargo, se puede calcular de esta forma una muy buena aproximación de la transformada de distancia euclidiana.

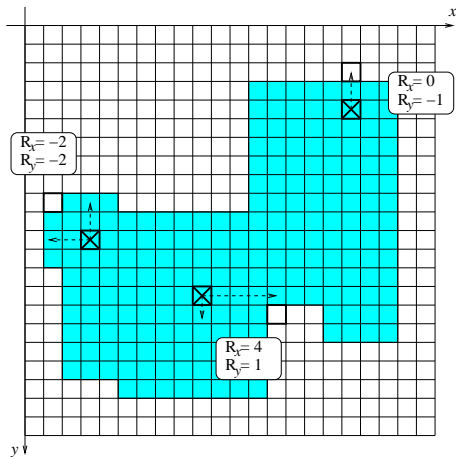
El algoritmo de Danielsson-Leymarie (DL) consiste en calcular, por cada píxel (x, y) de X , las coordenadas relativas $(R_x(x, y), R_y(x, y))$ del píxel de contorno mas cercano, es decir que el píxel de X^c lo mas cerca de (x, y) habra coordenadas $(x + R_x(x, y), y + R_y(x, y))$.

La transformada de distancia euclidiana de (x, y) vale entonces:

$$F_E^X(x, y) = \sqrt{R_x(x, y)^2 + R_y(x, y)^2}$$

Algoritmo de Danielsson-Leymarie

El algoritmo DL propaga, por cálculos recursivos, los valores de las coordenadas relativas (R_x, R_y) .



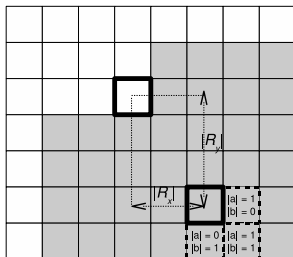
Algoritmo de Danielsson-Leymarie

El cuadrado de la transformada de distancia euclidiana es calculado por sumación marginal: *cuando un número z aumenta de 1, su cuadrado aumenta de $2z + 1$.*

$$(|R_x| + |a|)^2 + (|R_y| + |b|)^2 = R_x^2 + R_y^2 + 2|aR_x| + 2|bR_y| + a^2 + b^2$$

Entonces:

$$F_E^X(x+a, y+b)^2 = F_E^X(x, y)^2 + 2|aR_x| + 2|bR_y| + a^2 + b^2$$



Algoritmo de Danielsson-Leymarie

Notaciones: Se denota la *vecindad causal* $V^- = \{(-1, -1), (0, -1), (+1, -1), (-1, 0)\}$, y la *vecindad anticausal* $V^+ = \{(+1, 0), (-1, +1), (0, +1), (+1, +1)\}$.

Se denota $\Delta f^{(a,b)}(x, y) = 2(|aR_x(x + a, y + b)| + |bR_y(x + a, y + b)|) + a^2 + b^2$ el aumentación marginal dal cuadrado de la transformada de distancia euclidiana al pasar del punto $(x + a, y + b)$ al punto (x, y) .

Algoritmo de Danielsson-Leymarie

Inicialización

por cada píxel (x, y) :

si $(x, y) \notin X : \{f(x, y) = 0; R_x(x, y) = 0; R_y(x, y) = 0;\}$

si $(x, y) \in X : \{f(x, y) = \infty; R_x(x, y) = 0; R_y(x, y) = 0;\}$

Pasada causal

por y de 0 a H :

por x de 0 a W :

$$(1) (a, b) = \arg \min_{(u,v) \in V^-} [f(x + u, y + v) + \Delta f^{(u,v)}(x, y)]$$

$$(2) R_x(x, y) = R_x(x + a, y + b) + a$$

$$R_y(x, y) = R_y(x + a, y + b) + b$$

$$(3) f(x, y) = f(x + a, y + b) + \Delta f^{(a,b)}(x, y)$$

Algoritmo de Danielsson-Leymarie

Pasada anticausal

por y de H a 0 :

por x de W a 0 :

$$(1) (a, b) = \arg \min_{(u,v) \in V^+} [f(x + u, y + v) + \Delta f^{(u,v)}(x, y)]$$

$$(2) R_x(x, y) = R_x(x + a, y + b) + a$$

$$R_y(x, y) = R_y(x + a, y + b) + b$$

$$(3) f(x, y) = f(x + a, y + b) + \Delta f^{(a,b)}(x, y)$$

Algoritmo de Danielsson-Leymarie

En realidad, las 2 pasadas producen errores en los valores de la transformada de distancia, que pueden ser corregidos por pasadas suplementales:

transformada de distancia obtenida después de 2 pasadas:

5	4	5	8	13	20	29	40	53	74	89
2	1	2	5	10	17	26	37	52	65	80
1	0	1	4	9	16	25	34	45	58	73
1	0	1	4	8	13	20	29	40	53	68
1	0	1	2	5	10	17	26	37	50	65
2	1	0	1	4	9	16	25	36	49	64
4	1	0	1	4	9	16	25	36	49	64
4	1	0	1	4	9	16	25	36	49	64

El valor «52» del píxel está atribuida por referencia al vecino con valor «34» :

$34 = 25 + 9(R_x = -5; R_y = +3)$, y entonces $52 = 36 + 16(R_x = -5 - 1; R_y = +3 + 1)$, por qué el valor «37» solo aparece después de la segunda pasada (anticausal).

Algoritmo de Danielsson-Leymarie

2 pasadas suplementales aumentan la precisión:

transformada de distancia obtenida después de 4 pasadas:

5	4	5	8	13	20	29	40	53	68	85
2	1	2	5	10	17	26	37	50	65	80
1	0	1	4	9	16	25	34	45	58	73
1	0	1	4	8	13	20	29	40	53	68
1	0	1	2	5	10	17	26	37	50	65
2	1	0	1	4	9	16	25	36	49	64
4	1	0	1	4	9	16	25	36	49	64
4	1	0	1	4	9	16	25	36	49	64

El valor «50» del píxel está atribuida por referencia al vecino con valor «37» :

$$37 = 36 + 1(R_x = -6; R_y = +1)$$

$$y \text{ entonces } 50 = 49 + 1(R_x = -6 - 1; R_y = +1 + 0)$$

Algoritmo de Danielsson-Leymarie

Pasada causal ; con retorno cada línea

por y de 0 a H :

por x de 0 a W :

$$(1) (a, b) = \arg \min_{(u,v) \in V^-} [f(x+u, y+v) + \Delta f^{(u,v)}(x, y)]$$

$$(2) R_x(x, y) = R_x(x+a, y+b) + a$$

$$R_y(x, y) = R_y(x+a, y+b) + b$$

$$(3) f(x, y) = f(x+a, y+b) + \Delta f^{(a,b)}(x, y)$$

por x de W a 0 :

Si $f(x+1, y) + \Delta f^{(1,0)}(x, y) < f(x, y)$:

$$R_x(x, y) = R_y(x+1, y) + 1 ; R_y(x, y) = R_y(x+1, y)$$

$$f(x, y) = f(x+1, y) + \Delta f^{(1,0)}(x, y)$$

Algoritmo de Danielsson-Leymarie

Pasada anticausal ; con retorno cada línea

por y de H a 0 :

por x de W a 0 :

$$(1) (a, b) = \arg \min_{(u,v) \in V^+} [f(x + u, y + v) + \Delta f^{(u,v)}(x, y)]$$

$$(2) R_x(x, y) = R_x(x + a, y + b) + a$$

$$R_y(x, y) = R_y(x + a, y + b) + b$$

$$(3) f(x, y) = f(x + a, y + b) + \Delta f^{(a,b)}(x, y)$$

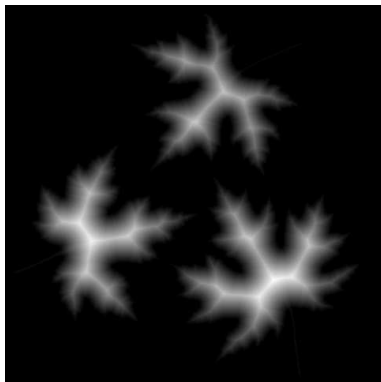
por x de 0 a W :

Si $f(x - 1, y) + \Delta f^{(-1,0)}(x, y) < f(x, y)$:

$$R_x(x, y) = R_y(x - 1, y) - 1 ; R_y(x, y) = R_y(x - 1, y)$$

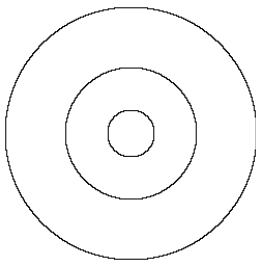
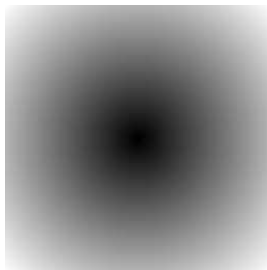
$$f(x, y) = f(x - 1, y) + \Delta f^{(-1,0)}(x, y)$$

Transformada de distancia euclidiana: ilustración



Transformada de distancia euclidiana calculada con el algoritmo DL (después del cálculo de la raíz cuadrada).

Transformada de distancia euclidiana: ilustración



Transformada de distancia del complementario del píxel central, y círculos discretos (radios 25, 75 et 125), obtenidos con el algoritmo DL.

Índice

- 1 Formalismo, Definiciones y Notaciones
 - Teselaciones y Imágenes discretas
 - Topologías en la malla cuadrada
 - Distancias en \mathbb{Z}^n
- 2 Transformadas de distancia: Algoritmos
 - Algoritmos básicos
 - Distancias casi euclidianas
- 3 Aplicaciones de las transformadas de distancia
 - Operadores morfológicos
 - Esqueleto morfológico y erosión última
 - Esqueletos conexos multi escala
- 4 Conclusión

Erosión y dilatación

El *erosión* y la *dilatación* son los operadores fundamentales del procesamiento morfológico de las imágenes. Sea $X \subset \mathbb{Z}^n$ una imagen binaria ; $B \subset \mathbb{Z}^n$ un *elemento estructurante*.

Erosión morfológica

$$\varepsilon_B(X) = \bigcap_{b \in B} X_{-b} = \{ z \in \mathbb{Z}^n; \forall b \in B, \exists x \in X : z = x - b \}$$

Dilatación morfológica

$$\delta_B(X) = \bigcup_{b \in B} X_{-b} = \{ z \in \mathbb{Z}^n; \exists b \in B, \exists x \in X : z = x - b \}$$

Erosión y dilatación

Cuando el elemento estructurante B es una bola de la distancia d :
 $B_\lambda(x) = \{y \in \mathbb{Z}^n; d(x, y) \leq \lambda\}$, el erosión y la dilatación se calculan por *umbrales* de la transformada de distancia:

Erosión por una bola

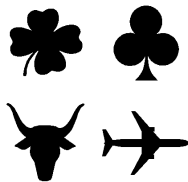
$$\varepsilon_{B_\lambda}(X) = \{z \in \mathbb{Z}^n; d(z, X^c) \geq \lambda\}$$

Dilatación por una bola

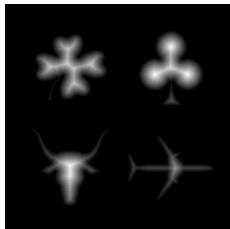
$$\delta_{B_\lambda}(X) = \{z \in \mathbb{Z}^n; d(z, X) < \lambda\}$$

Erosión morfológica

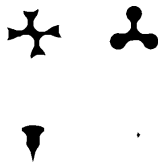
Erosión por una bola euclidiana de radio $\lambda = 20$:



X



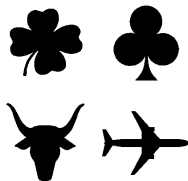
F_E^X



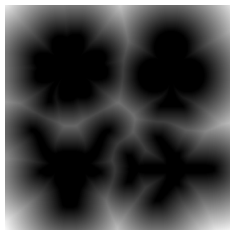
$\{z; F_E^X(z) \geq 20\}$

Dilatación morfológica

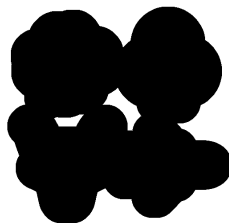
Dilatación por una bola euclidiana de radio $\lambda = 50$:



X



$F_E^{X^c}$



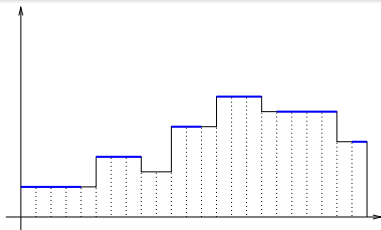
$\{z; F_E^{X^c}(z) < 50\}$

Máximos locales y máximos regionales

Máximos locales

Los *máximos locales* de una imagen digital F son los píxeles p cuyo valor $F(p)$ es superior o igual a los de sus vecinos.

Máximos locales de una función digital 1d (azul).



Cálculo de los máximos locales

Los máximos locales se calculan *de manera local*. Son asociados a una topología: 4-conexa o 8-conexa :

Máximos locales 4-conexa

$$m_4(F) = \{z \in \mathbb{Z}^2; \forall q, d_4(q, z) = 1 : F(z) \geq F(q)\}$$

Máximos locales 8-conexa

$$m_8(F) = \{z \in \mathbb{Z}^2; \forall q, d_8(q, z) = 1 : F(z) \geq F(q)\}$$

Cálculo de los máximos regionales 1/2

Los máximos regionales se calculan *de manera no local*. Por ejemplo, aquí está el algoritmo de cálculo de los máximos regionales 4-conexos.

(1) Inicialización de una cola (FIFO): máximos estrictos (i.e bordes de los máximos regionales)

por y de 0 a H :

por x de 0 a W :

Si $\forall (a, b), |a| + |b| = 1, F(x, y) > F(x + a, y + b)$:

$$M_4 = M_4 \cup \{(x, y)\}$$

$$L = L \cup \{(x, y)\}$$

Cálculo de los máximos regionales 2/2

(2) Exploración de la cola: propagación de los máximos

por cada $(x, y) \in L$:

por cada $(a, b), |a| + |b| = 1, F(x, y) = F(x + a, y + b)$:

Si $(x + a, y + b) \notin M_4$:

$$M_4 = M_4 \cup \{(x + a, y + b)\}$$

$$L = L \cup \{(x + a, y + b)\}$$

$$L = L \setminus \{(x, y)\}$$

Al final, M_4 contiene los máximos regionales de F , en el sentido de la 4-conectividad.

Máximos locales y esqueletos morfológicos

Los máximos locales de la transformada de distancia coinciden con el conjunto de los centros de bolas máximas, denominado *esqueletos morfológicos*

Esqueletos morfológicos

$$S_d(X) = \{z \in \mathbb{Z}^2; \exists n \in \mathbb{N}, B_d(z, n) \subset X, \forall (q, m) \in \mathbb{Z}^2 \times \mathbb{N}, B_d(z, n) \subset B_d(q, m) \subset X \Rightarrow (q, m) = (z, n)\}$$

Propiedad

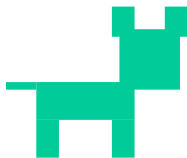
$$S_{d_4}(X) = m_4(F_4^X)$$
$$S_{d_8}(X) = m_8(F_8^X)$$

Máximos locales y esqueletos morfológicos

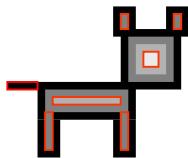
Propiedad

$$S_{d_4}(X) = m_4(F_4^X)$$

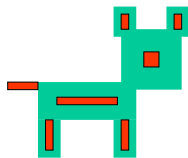
$$S_{d_8}(X) = m_8(F_8^X)$$



X



F_8^X

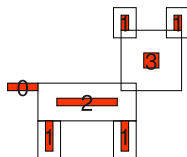
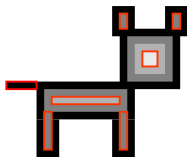
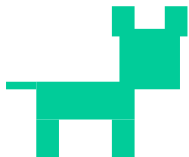


$S_{d_8}(X)$

Máximos locales y esqueletos morfológicos

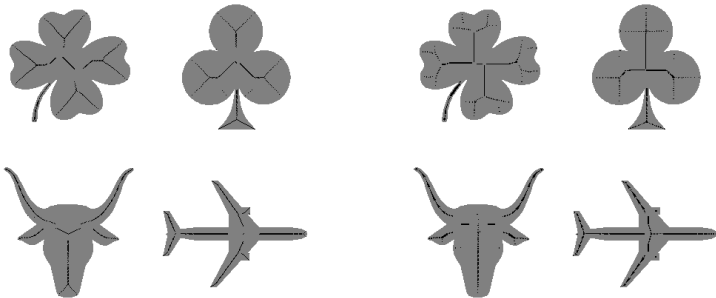
Aplicación : Código de una imagen binaria. El conocimiento de la transformada de distancia sobre el esqueleto morfológico provee una representación compacta de la imagen binaria:

$$X = \bigcup_{z \in S_d(X)} B(z, F_d^X(z))$$



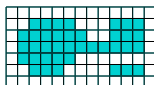
Esqueleto morfológico: ilustraciones

esqueleto morfológicos: negro (Imagen original X : gris).

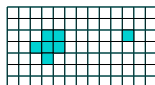

 $S_{d_4}(X)$
 $S_{d_8}(X)$

Máximos regionales y erosión última

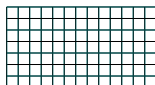
Los máximos regionales de la transformada de distancia d coinciden con el conjunto denominado *erodados últimos*, correspondiente a la unión de las componentes conexas que desaparecen enteramente bajo la acción de una secuencia de erosiones reiteradas por la bola de radio 1 de la distancia d .



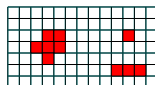
$$X_0 = X$$



$$X_1 = \varepsilon_{B_8^1}(X_0)$$



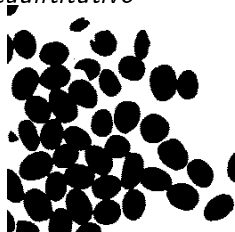
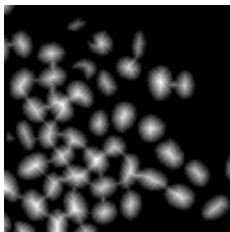
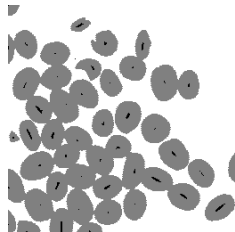
$$X_2 = \varepsilon_{B_8^1}(X_1)$$



$$M_8(X)$$

Máximos regionales y erosión última

Aplicación: La erosión última sirve a singularizar partículas que se recubren mutuamente. *Ejemplo: enumeración de células en análisis cuantitativo*


 X

 F_8^X

 $M_8(F_8^X)$ (negro)

Esqueletos euclidianos multi escala

Una de la mas potentes aplicaciones de las transformadas de distancia es el *esqueleto euclidiano multi escala*. El principio es el siguiente:

- 1 Asociar una *etiqueta* única a cada píxel de contorno.
- 2 *Propagar* el valor de las etiquetas a los píxeles mas cercanos.
- 3 Calcular una *función de choque* local según la diferencia de etiquetas entre píxeles adyacentes.
- 4 El *esqueleto* es obtenido por umbral de la función de choque.

Etiquetaje de contornos

Sea X una imagen binaria.

Contorno en 4-conectividad

$$\partial_X^4 = \{z \in X; \exists q, d_4(z, q) = 1, q \notin X\}$$

Contorno en 8-conectividad

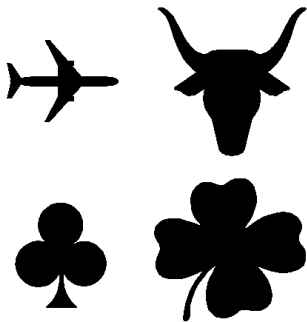
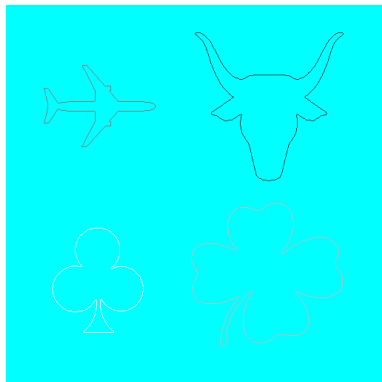
$$\partial_X^8 = \{z \in X; \exists q, d_4(z, q) = 1, q \notin X\}$$

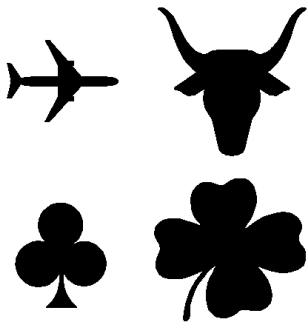
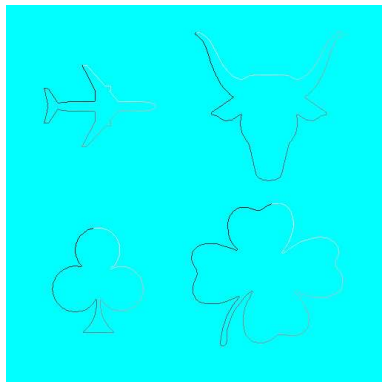
Observe: El contorno en 4-conectividad forma una curva cerrada 8-conexa por cada componente conexa de X . El contorno en 8-conectividad forma una curva cerrada 4-conexa por cada componente conexa de X .

Etiquetaje de contornos

Sea X una imagen binaria. Sea ∂_X el contorno de X . El etiquetaje de contornos de X consiste en atribuir una pareja de etiquetas (Λ, λ) a cada píxel de ∂_X , tal como:

- Λ identifica las componentes conexas de ∂_X .
- λ atribuye un número distinto a cada píxel de cada componente, según cierto trayecto (e.g. sentido trigonométrico).

Etiquetaje de contornos: Λ  X  Λ_X

Etiquetaje de contornos: λ  X  λX

Etiquetaje de contornos 1/2

Inicialización

por y de 0 a H :

por x de 0 a W :

Si $(x, y) \in \partial_X^4$:

$$\Lambda(x, y) = \infty$$

Indice = 0

Etiquetaje de contornos 2/2

Exploración de los contornos conexos

por y de 0 a H :

por x de 0 a W :

Si $\Lambda(x, y) = \infty$

Indice = Indice + 1 ; $\Lambda(x, y) = \text{Indice}$;

Numero = 1 ; $\lambda(x, y) = \text{Numero}$;

Mientras $\exists(x', y'), d_8((x, y), (x', y')) = 1, \Lambda(x', y') = \infty$

$\Lambda(x', y') = \text{Indice}$; Numero = Numero + 1 ;

$\lambda(x', y') = \text{Numero}$; $(x, y) = (x', y')$;

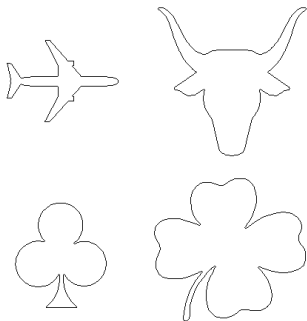
Propagación de las etiquetas

La propagación de las etiquetas a los píxeles mas cercanos se hace simplemente usando el algoritmo de cálculo de la función distancia d sobre el complementario del contorno $(\partial_X)^c$, asociando a cada píxel (x, y) las coordenadas relativas $(R_x(x, y), R_y(x, y))$ del píxel de ∂_X el más cerca de (x, y) .

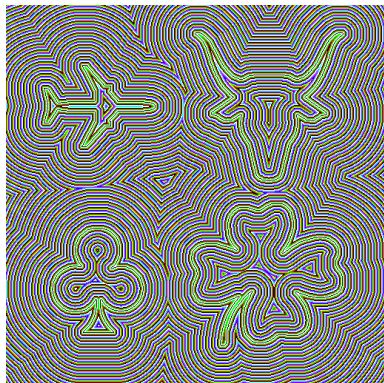
Si L es una función etiqueta sobre ∂_X , la propagación de la etiqueta L según la distancia d es la función definida sobre X como sigue:

$$\Pi_d^L(x, y) = L(x + R_x, y + R_y)$$

Propagación de las etiquetas: Ilustración



$$\partial^4 X$$

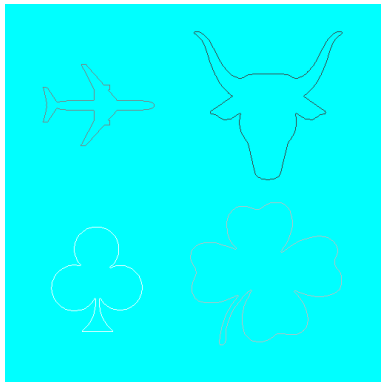


$$F_E^{(\partial x)^c}$$

Propagación de las etiquetas: SKIZ et esqueletos

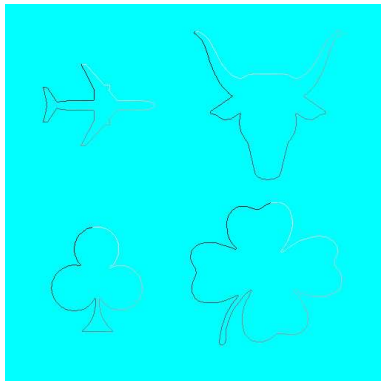
- La propagación de la etiqueta Λ (componente conexa) provee la partición de X en zonas de influencia (ou SKIZ).
- La propagación de la etiqueta λ (enumeración de contorno) calcula las zonas de influencia de cada píxel de ∂X , lo que por diferenciación, proveerá el esqueleto de X .

Propagación de las etiquetas Λ : SKIZ


 Λ_x

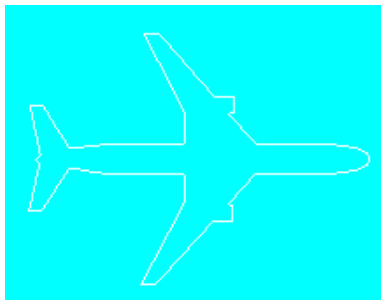
 $\prod_E^{\Lambda_x}$

Propagación de las etiquetas λ

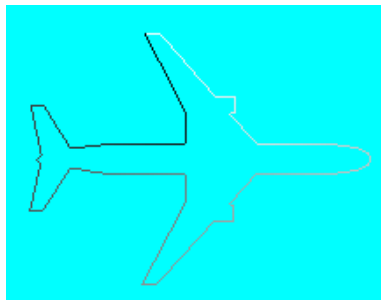

 λ_x

 $\prod_E^{\lambda_x}$

Propagación de las etiquetas: detalle

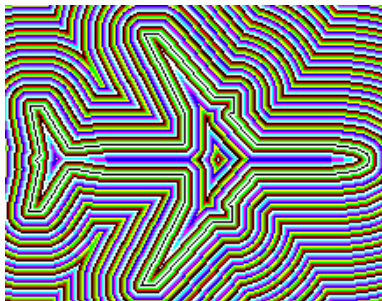


Λ_X (1 etiqueta)



λ_X (679 etiquetas)

Propagación de las etiquetas: detalle



$$F_E^{(\partial x)^c}$$



$$\Pi_E^{\lambda x}$$

Función de choque

- La función de choque asocia a cada píxel p un valor proporcional a la «lejanía» máxima entre el píxel de contorno correspondiente a la etiqueta de p y los que corresponden a la etiqueta de los píxeles adyacentes a p .
- La lejanía es asociada a una función de costo κ sobre $\partial_X \times \partial_X$, donde cada píxel del contorno ∂_X está identificado por su pareja de etiquetas (λ_X, λ_X) .

Se denota $\mathcal{N}_X(p) = (p + R_x(p), p + R_y(p))$, i.e. el punto de ∂_X el más cerca de p .

Función de choque

Función de choque 8-conexa

$$S_8(p) = \max_{d_4(p,q)=1} \kappa(\mathcal{N}_X(p), \mathcal{N}_X(q))$$

Función de choque 4-conexa

$$S_4(p) = \max_{d_8(p,q)=1} \kappa(\mathcal{N}_X(p), \mathcal{N}_X(q))$$

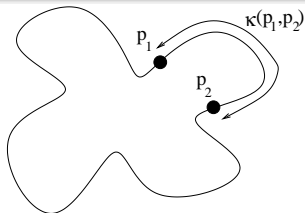
Observe la dualidad: se calcula el valor máximo en la 4-vecindad para un esqueleto 8-conexo, y recíprocamente.

Función de choque

La función de costo κ asociada es la *distancia geodésica* entre los dos píxeles $p_1 = \mathcal{N}_X(p)$ y $p_2 = \mathcal{N}_X(q)$ al largo del contorno ∂_X :

Función de costo «distancia geodésica»

$$\kappa(p_1, p_2) = d_{\partial_X}(p_1, p_2)$$

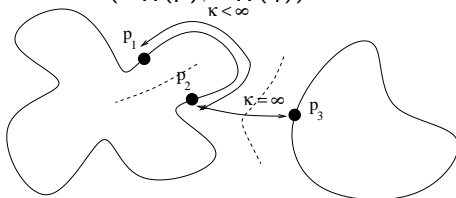


Función de choque

La función de choque se calcula muy simplemente comparando las etiquetas Λ_X y λ_X de los píxeles adyacentes:

(1) Si $\Lambda_X(p) \neq \Lambda_X(q)$, entonces p está a la frontera de una zona de influencia de un contorno conexa, y:

$$\kappa(\mathcal{N}_X(p), \mathcal{N}_X(q)) = \infty$$



Función de choque

(2) Si $\Lambda_X(p) = \Lambda_X(q)$ entonces la función de costo es igual a la diferencia de las etiquetas λ_X , módulo el número total de píxeles del contorno:

Función de costo simétrica

$$\kappa(\mathcal{N}_X(p), \mathcal{N}_X(q)) = |\Pi_E^{\lambda_X}(p) - \Pi_E^{\lambda_X}(q)| \pmod{|\partial_X|}$$

Función de costo asimétrica

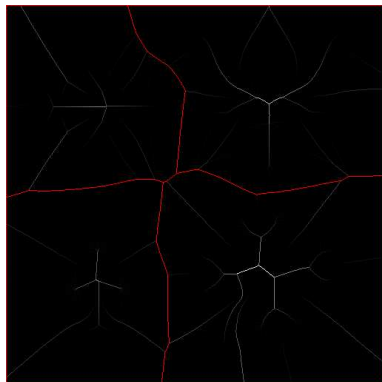
$$\kappa(\mathcal{N}_X(p), \mathcal{N}_X(q)) = \Pi_E^{\lambda_X}(p) - \Pi_E^{\lambda_X}(q) \pmod{|\partial_X|}$$

La función de costo simétrica produce un esqueleto centrado pero de espesor 2, la función de costo asimétrica produce un esqueleto de espesor 1, con un error posible de ubicación de medio píxel.

Función de choque



$$\Pi_E^{\lambda x}$$



$$S_8(\Pi_E^{\lambda x})$$

Esqueleto multi escala

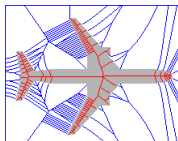
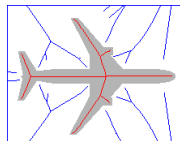
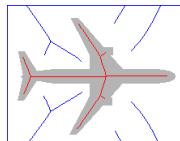
Una vez la función de choque S definida, el esqueleto de la escala σ está definido como el umbral de la función S al valor σ :

Esqueleto de la escala σ

$$Sk_{\sigma}(X) = \{z; S(z) \geq \sigma\}$$

Conectividad de los esqueletos multi escala

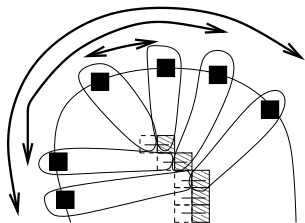
Propiedad: La función de choque asociada a la distancia geodésica al largo del contorno es *conexa-monótona*, es decir que por todo número entero n , el conjunto de los píxeles cuya función de choque es superior a n forma el mismo número de componentes conexas que la imagen X inicial.


 S_8

 $\{z; S_8(z) > 1\}$

 $\{z; S_8(z) > 5\}$

 $\{z; S_8(z) > 20\}$

Conectividad de los esqueletos multi escala

La propiedad de conexas-monotonía de la función de choque es debida al hecho que las zonas de influencia de los píxeles del contorno son *conexas*.

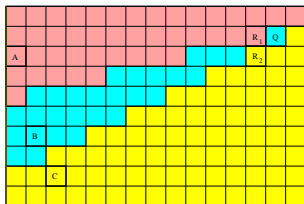
La conectividad de las zonas de influencia de los píxeles implica el crecimiento de la función de choque al largo de las curvas del esqueleto a partir de las extremidades:



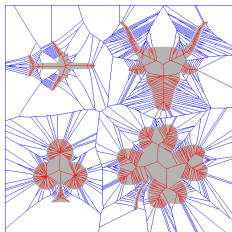
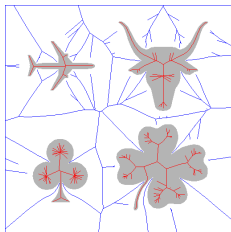
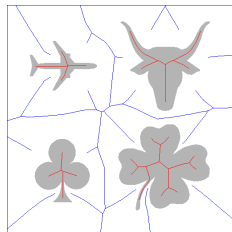
Conectividad de los esqueletos multi escala

La conectividad de las zonas de influencia también es una condición necesaria de conectividad de los esqueletos multi escala. Pues el algoritmo DL es *más adecuado* que una distancia euclidiana exacta para el cálculo de los esqueletos conexos.

Basándose sobre las distancias euclidianas exactas, se puede construir un camino conexo vinculando A, B y C, que tendrá un esqueleto de conectado:



Esqueleto multi escala


 $Sk_1(X)$

 $Sk_5(X)$

 $Sk_{50}(X)$

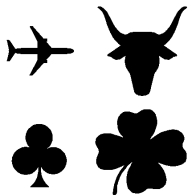
Reconstrucción multi escala

La reconstrucción de la escala σ está obtenida por la fórmula de inversión del esqueleto:

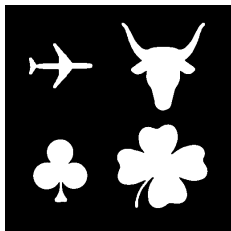
Reconstrucción de la escala σ

$$R_{\sigma}(X) = \bigcup_{z \in Sk_{\sigma}(X)} B_z(F_X^E(z))$$

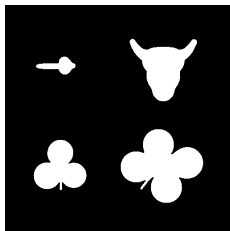
Reconstrucción multi escala



Escala 1



Escala 20



Escala 100

Mapa de reconstrucción

Todas las reconstrucciones multi escala pueden ser obtenidas muy rápidamente a partir de la mapa de reconstrucción definida por:

Mapa de reconstrucción

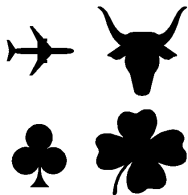
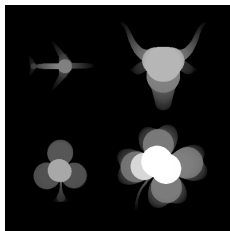
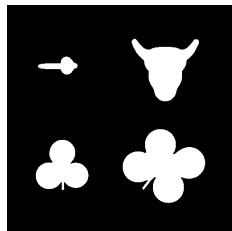
$$M_X(p) = \max_{z \in Sk_1(X); p \in B_z(F_X^E(z))} S_X(z)$$

Pues la reconstrucción de la escala σ se calcula por simple umbral de la función M_X :

Reconstrucción de la escala σ

$$R_\sigma(X) = \{z; M_X(z) \geq \sigma\}$$

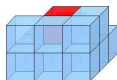
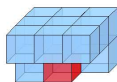
Mapa de reconstrucción


 X

 M_X

 $\{z; M_X(z) > 100\}$

¿ Extensión 3d o n-d ?

El cálculo de las transformadas de distancia se generalizan fácilmente a las dimensiones superiores:

- 1 Adaptando la vecindad de cálculo :



V^- en 3d (14 vóxeles) V^+ en 3d (14 vóxeles)

- 2 Modificando la función de incremento $\Delta f^a(\mathbf{x})$:

$$F_E^X(\mathbf{x} + \mathbf{a}) = F_E^X(\mathbf{x}) + \Delta f^a(\mathbf{x}) = F_E^X(\mathbf{x}) + \sum_{i=1}^n 2|a_i R_{x_i}| + a_i^2$$

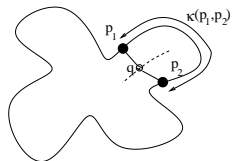
con $\mathbf{x} = (x_1, \dots, x_n)$, $\mathbf{a} = (a_1, \dots, a_n)$.

¿ Extensión 3d o n-d ?

En cambio, la función de choque usada en 2d no sirve en 3d:

En 3d, la distancia geodésica al largo del contorno entre 2 puntos del contorno ya no puede calcularse con simple diferencia de las etiquetas, como en 2d: $S(q) =$

$$\kappa(p_1, p_2) = d_{\partial X}(p_1, p_2)$$

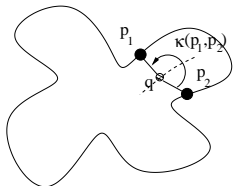


¿ Extensión 3d o n-d ?

Por esta razón, se usará en 3d una función de choque más fácilmente calculable:

Se puede definir otra función de choque a partir de la medida del ángulo entre los 2 puntos de contorno más cercanos:

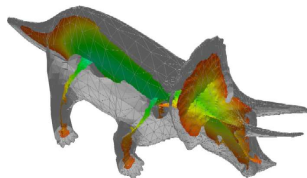
$$S(q) = \kappa(p_1, p_2) = \widehat{p_1 q p_2}$$



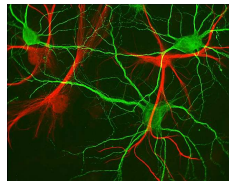
Observe: ¡ pero la preservación de la topología ya no está asegurada !

Aplicaciones

- Código, compresión, síntesis de patrones
- Identificación de modelos geométricos, topológicos.
- Morfometría.



(from A. Sud et al, Univ. North Carolina)



(from P. de Koninck, Univ. Laval)

Índice

- 1 Formalismo, Definiciones y Notaciones
 - Teselaciones y Imágenes discretas
 - Topologías en la malla cuadrada
 - Distancias en \mathbb{Z}^n
- 2 Transformadas de distancia: Algoritmos
 - Algoritmos básicos
 - Distancias casi euclidianas
- 3 Aplicaciones de las transformadas de distancia
 - Operadores morfológicos
 - Esqueleto morfológico y erosión última
 - Esqueletos conexos multi escala
- 4 Conclusión

Conclusión

- Transformada de distancia: herramienta potente para el procesamiento, el análisis y la síntesis de patrones.
- algoritmos eficaces: complejidad lineal, extensión multi dimensional.
- Perspectivas: teselaciones irregulares, grafos cualesquiera.

Bibliografía



ROSENFELD, A. AND PFLATZ, J.

Distance functions on digital pictures

Pattern Recognition 1 (1), 33-61. (1968)



DANIELSSON, P.-E.

Euclidean distance mapping.

Computer Graphics and Image Processing. 14, 227-248. (1980)



SERRA, J.

Image Analysis and Mathematical Morphology - Vol. I

Academic Press, London. (1982)

Bibliografía



BORGEFORS, G.

Distance transformations in digital images.

Computer Vision, Graphics, and Image Processing, 34, 344-371.
(1986)



LEYMARIE, F. AND LEVINE, M. D.

Fast raster scan distance propagation on the discrete rectangular lattice.

Computer Vision and Image Understanding 55, 1. (1992)



DA FONTOURA COSTA, L.

Robust Skeletonization through Exact Euclidean Distance Transform and its Application to Neuromorphometry.

Real-Time Imaging 6(6), 415-431. (2000)