Fonctions distances discrètes : Algorithmes et applications

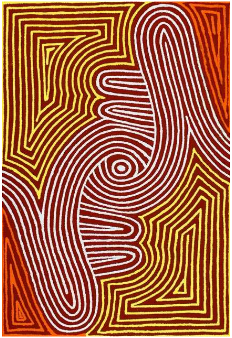
Antoine Manzanera
ENSTA
ParisTech

Objectifs du cours

- Introduction à la géométrie discrète.
  - Distance discrète
  - Topologie discrète
- Fonction distance discrète : Algorithmes
  - Distances liées à la topologie
  - Distances quasi-euclidiennes
- Fonction distance discrète : Applications
  - Analyse d'image : filtrage, segmentation
  - Représentation de formes : codage, reconnaissance

Problématique

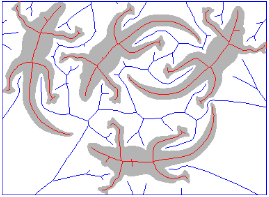
La fonction distance ou «Transformée en distance» est associée à un objet (ou ensemble, ou forme) X dans un espace E. Elle associe à chaque point de X sa distance au complémentaire de X dans E : $X^c = E \setminus X$. Cette fonction est bien sûr liée à une métrique d qu'on peut définir dans E.
On s'intéresse au calcul de cette fonction dans les espaces E discrets, et à son utilisation.



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

Applications

- Analyse d'images.
  - Filtrage : opérateurs morphologiques
  - Segmentation : Zones d'influence
  - Evolution d'interfaces
- Représentation de formes.
  - Squelettes
  - Codage, compression
  - Reconnaissance, indexation



Sommaire

Outline

Pavage de l'espace

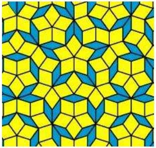
Un pavage est une *partition* de l'espace $\mathbb{R}^n$ en une famille de parties $\{P_i\}_{i \in I}$:

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

Les P_i s'appellent les *tesselles* (ou pixels, ou voxels).

Pajarita - Alhambra (Granada)

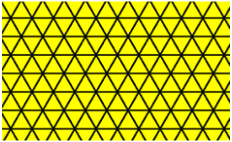
Pavage apériodique de Penrose



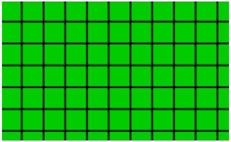
Pavage du plan

Dans le cas du plan $\mathbb{R}^2$, il n'existe que 3 *pavages réguliers*, c'est-à-dire respectant les contraintes suivantes :

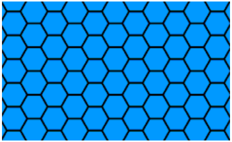
- les P_i sont tous identiques.
- les P_i sont des polygones réguliers (i.e. convexes, côtés égaux, angles égaux).
- chaque sommet de P_i est en contact avec d'autres sommets.



Pavage triangulaire



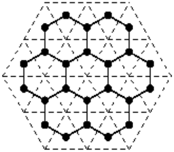
Pavage carré



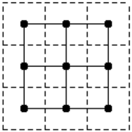
Pavage hexagonal

Pavage et maillage

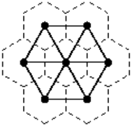
A tout pavage on peut associer un *graphe* dont les *sommets* représentent les tesselles, et dont les *arêtes* représentent la relation d'adjacence entre tesselles (2 tesselles sont *adjacentes* si elles ont un côté en commun). Un tel graphe est un *maillage* du plan.
Les pavages et les maillages réguliers sont des représentations duales du plan discret :



Maille hexagonale
(Pavage triangulaire)



Maille carrée
(Pavage carré)



Maille triangulaire
(Pavage hexagonal)

Propriétés des pavages réguliers

Le choix d'un type de pavage régulier en analyse d'images se fonde sur certaines propriétés du pavage :

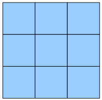
- Conformité à la géométrie du capteur.
- Récursivité (multi-résolution).
- Nombre de directions représentées.
- Extension aux dimensions supérieures.
- Représentation dans $\mathbb{Z}^n$.

La maille cubique

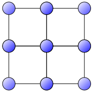
Pour toutes ces raisons, la maille carrée (en 2d), ou cubique (en 3d) est la plus utilisée en analyse d'image.

L'*espace discret* est alors représenté par $\mathbb{Z}^n$.

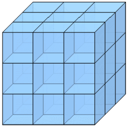
Un *pixel* (2d) ou *voxel* (3d) est un élément de $\mathbb{Z}^n$.



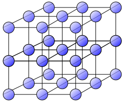
Pavage 2d



Maillage 2d



Pavage 3d



Maillage 3d

Image binaire

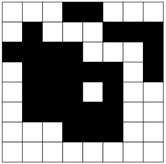
Soit $\mathbb{Z}^n$ l'espace discret.

Une image binaire I est définie comme un sous-ensemble de $\mathbb{Z}^n$:

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

Le pixel $p \in \mathbb{Z}^n$ sera représenté en noir si et seulement si $p \in I$.

Présentation «Pavage»



Présentation «Maillage»

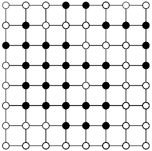


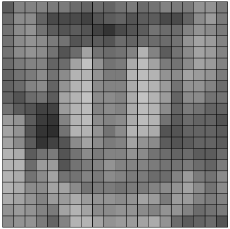
Image numérique

Soit $\mathbb{Z}^n$ l'espace discret.

Une image numérique F est définie comme une fonction de $\mathbb{Z}^n$ à valeurs entières :

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

Le pixel $p \in \mathbb{Z}^n$ sera représenté avec un niveau de gris proportionnel à $F(p)$.



Topologies dans la maille carrée

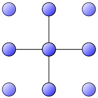
La topologie dans les images discrètes est construite à partir de la relation de connexité induite par le graphe du maillage (X, S), où X représente les sommets et S les arêtes.

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

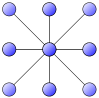
Soient x et y 2 points de X, par définition x et y sont *adjacents* si :

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

Dans la maille carrée, 2 types de relations d'adjacence peuvent être considérée :



4-connexité

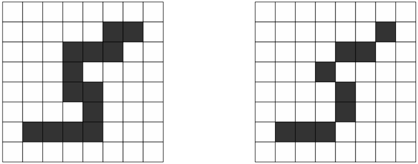


8-connexité

Chemins et composantes connexes

La clôture transitive de la relation d'adjacence $\approx$ est une relation d'équivalence : «il existe un *chemin connexe* entre x et y» :

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

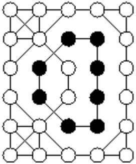


Chemin 4-connexe   Chemin 8-connexe

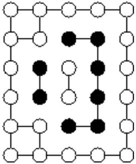
Les classes d'équivalences de la relation «$\sim$» s'appellent les *composantes connexes* de X.

Trous et Théorème de Jordan

Dans la maille carrée, la notion de trou dans un objet X ($X \subset \mathbb{Z}^2$), qui doit correspondre à une composante connexe finie du complémentaire X^c, n'est pas bien définie...



8-connexité

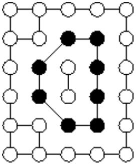


4-connexité

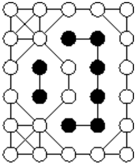
Ce problème est lié à la validité du théorème de Jordan, selon lequel une courbe simple fermée sépare le plan en 2 composantes connexes, dont une bornée.

Trous et Théorème de Jordan

...sauf si l'on considère différentes connexités pour X et pour X^c :



8-connexité



4-connexité

Le théorème de Jordan est vérifié pour la (8-4)-connexité et pour la (4,8)-connexité.

Trous et Théorème de Jordan

Exemple : *Combien y a-il de composantes connexes, et de trous, (1) pour la (8-4)-connexité, (2) pour la (4,8)-connexité ?*

Boules discrètes

Distance dans $\mathbb{Z}^n$

$d : \mathbb{Z}^n \times \mathbb{Z}^n$ est une *distance*

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

Boule discrète

Soit d une distance de $\mathbb{Z}^n$, $x \in \mathbb{Z}^n$, $r \in \mathbb{N}$.

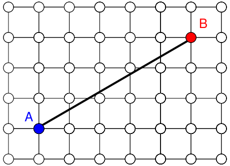
La *boule* de centre x et de rayon r est définie par :

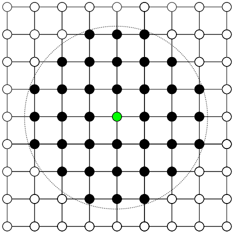
$B_r^d(x) = \{y \in \mathbb{Z}^n; d(x,y) \leq r\}$.

Distance euclidienne

Distance euclidienne dans $\mathbb{Z}^2$

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





$$d_E(A, B) = \sqrt{5^2 + 3^2} = \sqrt{34}$$

$$B^{d_E}_{\sqrt{10}}(C) = \{y \in \mathbb{Z}^n ; d_E(C, y) \leq \sqrt{10}\}$$

Distance d_4

Si l'on définit la distance entre 2 points x et y comme la longueur du plus court chemin connecté entre x et y, la topologie 4-connexe induit une distance d_4. En pondérant toutes les arêtes du maillage par la valeur 1, on trouve :

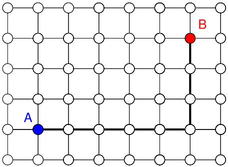
Distance d_4 dans $\mathbb{Z}^2$

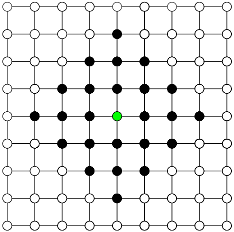
$$d_4(x, y) = |x_1 - y_1| + |x_2 - y_2|$$

Distance d_4

Distance d_4 dans $\mathbb{Z}^2$

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





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

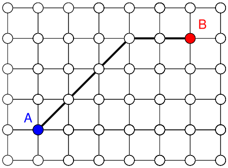
$B_3^{d_4}(C) = \{y \in \mathbb{Z}^n; d_4(C, y) \leq 3\}$

Distance d_8

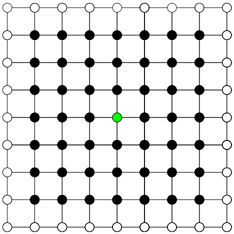
De même la topologie 8-connexe induit une distance d_8 définie par :

Distance d_8 dans $\mathbb{Z}^2$

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



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



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

Outline

1. Formalisme, Définitions et Notations
   - Pavages, Maillages et Images discrètes
   - Topologies dans la maille carrée
   - Distances dans $\mathbb{Z}^n$
2. Fonctions distances : Algorithmes
   - Algorithmes de base
   - Distances quasi-euclidiennes
3. Applications des fonctions distances
   - Opérateurs morphologiques
   - Squelette morphologique et érodés ultimes
   - Squelettes connexes multi-échelles
4. Conclusion

Fonction distance

Fonction distance

Soit d une distance dans $\mathbb{Z}^n$, $X \subset \mathbb{Z}^n$ une image binaire.
La *fonction distance* f_d^X est définie par :

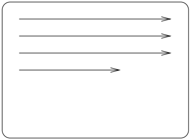
$$\begin{array}{rccc} f_d^X : & \mathbb{Z}^n & \longrightarrow & \mathbb{N} \\ & x & \longmapsto & d(x, X^c) \end{array}$$

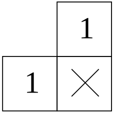
Fonction distance d_4

Soit X une image binaire contenue dans un support de dimension $W \times H$. La fonction distance d_4 sur X est calculée par l'algorithme récursif suivant, correspondant à 2 balayages successifs :

(1) Balayage 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

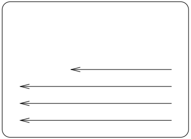


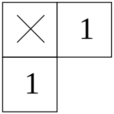


Fonction distance d_4

(2) Balayage anticausal

$$
\begin{array}{l}
\text{for j} = \text{H downto } 1 \\
\quad \text{for i} = \text{W downto } 1 \\
\quad\quad F_4^X(i,j) = \min(\; F_4^X(i,j), \\
\quad\quad\quad\quad\quad\quad\quad\quad F_4^X(i+1,j)+1, \\
\quad\quad\quad\quad\quad\quad\quad\quad F_4^X(i,j+1)+1); \\
\quad \text{endfor} \\
\text{endfor}
\end{array}
$$





La complexité de l'algorithme est de 2 comparaisons par pixel.

Fonction distance d_4 : illustration

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) Initialisation

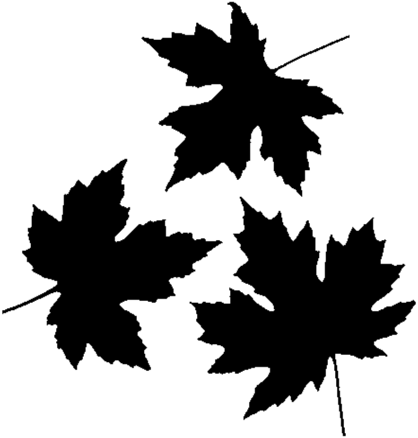
0	0	0	0	0	0	0	0
0	1	1	1	1	1	1	0
0	1	2	2	2	2	2	0
0	1	2	0	0	1	2	0
0	1	2	1	1	2	3	0
0	1	2	2	2	3	4	0
0	0	0	0	0	0	0	0

(1) Après balayage causal

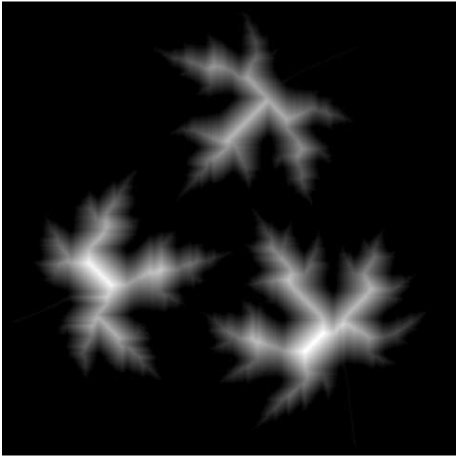
0	0	0	0	0	0	0	0
0	1	1	1	1	1	1	0
0	1	2	1	1	2	1	0
0	1	1	0	0	1	1	0
0	1	2	1	1	2	1	0
0	1	1	1	1	1	1	0
0	0	0	0	0	0	0	0

(2) Après 2 balayages

Fonction distance d_4 : illustration

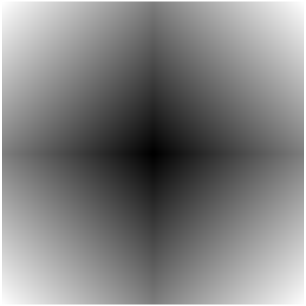


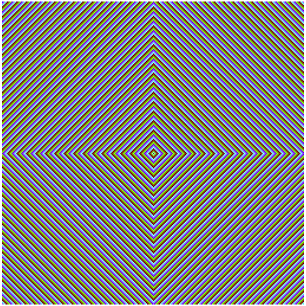
Fonction distance d_4 : illustration



Fonction distance d_4 : illustration

A gauche : fonction distance d_4 du complémentaire d'un ensemble réduit à un pixel (centre de l'image).
A droite : lignes de niveaux r (en couleur) de la fonction distance : cercles discrets de rayon r.



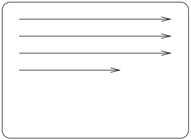


Fonction distance d_8

De même, la fonction distance d_8 sur une image binaire X se calcule de la façon suivante :

(1) Balayage 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	×	

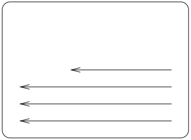
Fonction distance d_8

(2) Balayage anticausal

for j = H downto 1
    for i = W downto 1

$$
\begin{aligned}
F_8^X(i,j) = \min(\ & F_8^X(i,j),\\
& F_8^X(i+1,j+1)+1,\\
& F_8^X(i,j+1)+1,\\
& F_8^X(i-1,j+1)+1,\\
& F_8^X(i+1,j)+1);
\end{aligned}
$$

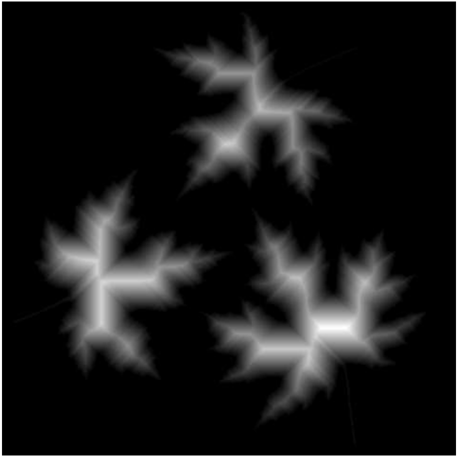
    endfor
endfor



	×	1
1	1	1

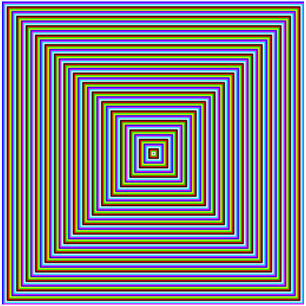
La complexité de l'algorithme est de 4 comparaisons par pixel.

Fonction distance d_8 : illustration

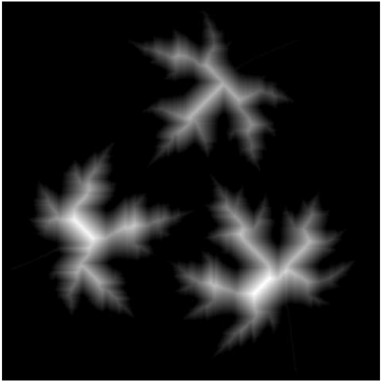


Fonction distance d_8 : illustration

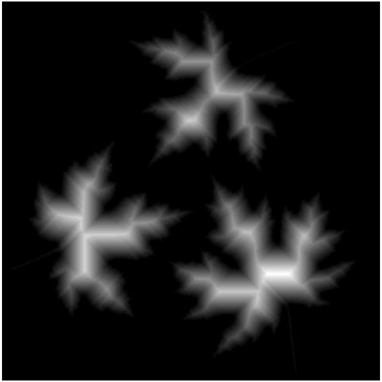
A gauche : fonction distance d_8 du complémentaire d'un ensemble réduit à un pixel (centre de l'image).
A droite : lignes de niveaux r (en couleur) de la fonction distance : cercles discrets de rayon r.



Comparaison distances d_4 et d_8



Fonction distance d_4



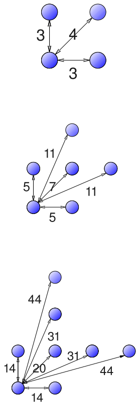
Fonction distance d_8

Distances de chanfrein

Les distances de chanfrein sont une tentative pour limiter le caractère anisotropique des distances d_4 et d_8, et de se rapprocher de la distance euclidienne.

Le principe consiste encore à définir la distance comme la longueur du plus court chemin entre deux points, mais en pondérant différemment les arêtes selon leur type. Par exemple, la distance du chanfrein d_{3-4} (*en haut*)

On peut également travailler sur des maillages plus complexes, i.e. des voisinages plus grands : d_{5-7-11} (*au milieu*), ou encore $d_{14-20-31-44}$ (*en bas*).



Fonction distance d_{3-4}

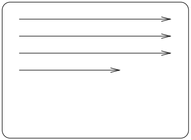
Par exemple, la fonction distance d_{3-4} sur une image binaire X se calcule de la façon suivante :

(1) Balayage causal

for j = 1 to H
    for i = 1 to W
        if $(i,j) \in X$: $F^X_{3-4}(i,j) = \infty$;
        else $F^X_{3-4}(i,j) = 0$;

$$
\begin{aligned}
F^X_{3-4}(i,j) = \min(\ & F^X_{3-4}(i,j), \\
& F^X_{3-4}(i-1,j-1)+4, \\
& F^X_{3-4}(i,j-1)+3, \\
& F^X_{3-4}(i+1,j-1)+4, \\
& F^X_{3-4}(i-1,j)+3);
\end{aligned}
$$

    endfor
endfor

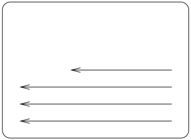


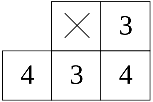
4	3	4
3		

Fonction distance d_{3-4}

(2) Balayage anticausal

for j = H downto 1
    for i = W downto 1
        $F^X_{3-4}(i,j) = \frac{1}{3} \times \min(F^X_{3-4}(i,j),$
            $F^X_{3-4}(i+1,j+1)+4,$
            $F^X_{3-4}(i,j+1)+3,$
            $F^X_{3-4}(i-1,j+1)+4,$
            $F^X_{3-4}(i+1,j)+3);$
    endfor
endfor

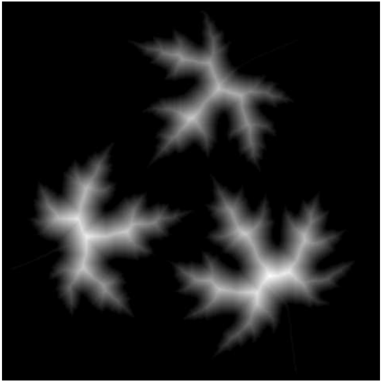




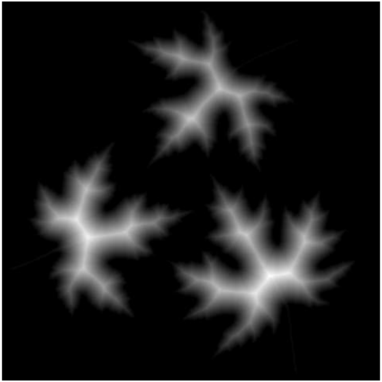
La complexité de l'algorithme est de 4 comparaisons par pixel.

Noter la division par 3 pour normaliser.

Distances de chamfrein : illustrations



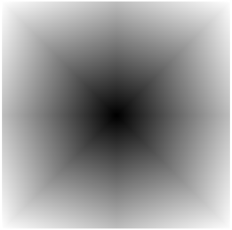
Fonction distance d_{3-4}



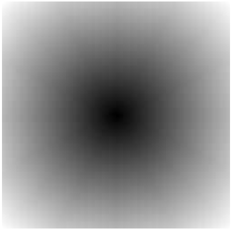
Fonction distance d_{5-7-11}

Distances de chamfrein : illustrations

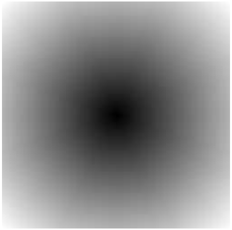
4	3	4
3	×	



	11		11	
11	7	5	7	11
	5	×		

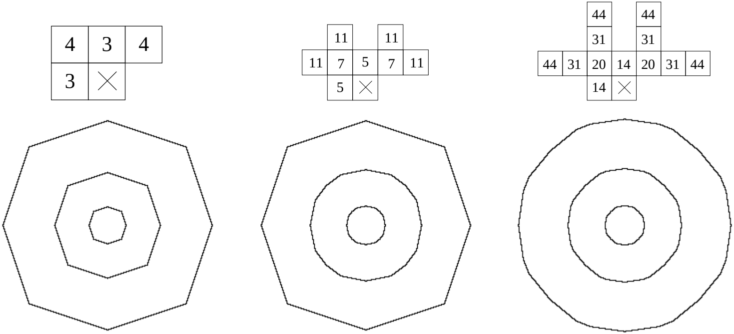


		44		44		
		31		31		
44	31	20	14	20	31	44
		14	×			



Masques de calcul (causal) et fonction distance du complémentaire du pixel central, pour les 3 premières distances de chamfrein.

Distances de chamfrein : illustrations

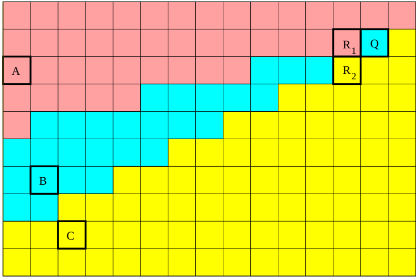


Masques de calcul (causal) et cercles discrets (rayons 25, 75 et 125), pour les 3 premières distances de chamfrein.

Fonction distance euclidienne

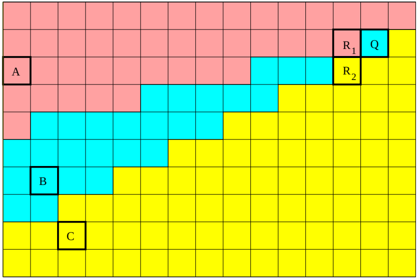
Problème : Peut-on calculer une fonction distance euclidienne *exacte* sur n'importe quelle image binaire X par une suite de balayages faisant intervenir uniquement des calculs dans un voisinage 3×3 ?

Réponse : NON !



Fonction distance euclidienne

$d_E(R_1, A) < d_E(R_1, B)$,
$d_E(R_2, C) < d_E(R_2, B)$ mais
$d_E(Q, B) < d_E(Q, A)$ et
$d_E(Q, B) < d_E(Q, C)$.
On ne peut donc pas décider localement de la fonction distance au point Q.



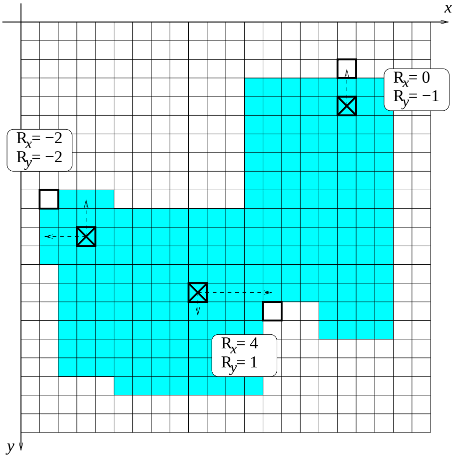
Algorithme de Danielsson-Leymarie

Cependant, on peut calculer de cette façon une très bonne approximation de la fonction distance euclidienne.
L'algorithme de Danielsson-Leymarie (DL) consiste à calculer, pour chaque pixel (x, y) de X, les coordonnées relatives $(R_x(x, y), R_y(x, y))$ du pixel de contour le plus proche, c'est-à-dire que le pixel de X^c le plus proche de (x, y) a pour coordonnées $(x + R_x(x, y), y + R_y(x, y))$.
La fonction distance euclidienne de (x, y) vaut alors :

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

Algorithme de Danielsson-Leymarie

L'algorithme DL propage, par des calculs récursifs, les valeurs des fonctions de coordonnées relatives (R_x, R_y).

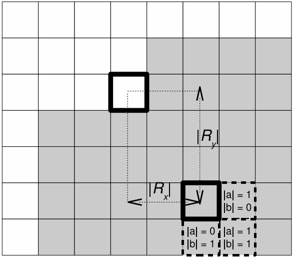


Algorithme de Danielsson-Leymarie

Le carré de la fonction distance euclidienne est calculé par sommation marginale : *quand un nombre z augmente de* 1*, son carré augmente de* $2z + 1$.

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

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



Algorithme de Danielsson-Leymarie

Notations : On note $V^- = \{(-1,-1),(0,-1),(+1,-1),(-1,0)\}$ le *voisinage causal*, et $V^+ = \{(+1,0),(-1,+1),(0,+1),(+1,+1)\}$ le *voisinage anti-causal*.
On note $\Delta f^{(a,b)}(x,y) = 2(|aR_x(x+a,y+b)|+|bR_y(x+a,y+b)|)+a^2+b^2$ l'augmentation marginale du carré de la fonction distance euclidienne quand on passe du point $(x+a,y+b)$ au point (x,y).

Algorithme de Danielsson-Leymarie

Initialisation

pour tout pixel (x, y):

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

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

Balayage direct

pour y de 0 à H :

pour x de 0 à 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) $f(x, y) = f(x + a, y + b) + \Delta f^{(a,b)}(x, y)$

Algorithme de Danielsson-Leymarie

Balayage rétrograde

pour y de H à 0 :

pour x de W à 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) $f(x,y) = f(x+a, y+b) + \Delta f^{(a,b)}(x,y)$

Algorithme de Danielsson-Leymarie

En réalité, les 2 balayages produisent des valeurs erronées de la fonction distance, qui peuvent être corrigées par des balayages supplémentaires :

Fonction distance obtenue après 2 scans :

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

La valeur «52» du pixel est attribuée par référence au voisin ayant la valeur «34» :

$34 = 25 + 9 (R_x = -5; R_y = +3)$, et donc $52 = 36 + 16 (R_x = -5 - 1; R_y = +3 + 1)$,

car la valeur «37» n'apparaît qu'au deuxième balayage (anticausal).

Algorithme de Danielsson-Leymarie

2 balayages supplémentaires augmentent la précision :

Fonction distance obtenue après 4 scans :

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

La valeur «50» du pixel est attribuée par référence au voisin ayant la valeur «37» :
$37 = 36 + 1(R_x = -6; R_y = +1)$
et donc $50 = 49 + 1(R_x = -6 - 1; R_y = +1 + 0)$

Algorithme de Danielsson-Leymarie

Balayage direct ; avec retour pour chaque ligne

pour y de 0 à H :

pour x de 0 à 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) $f(x,y) = f(x+a,y+b) + \Delta f^{(a,b)}(x,y)$

pour x de W à 0 :

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

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

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

Algorithme de Danielsson-Leymarie

Balayage rétrograde ; avec retour pour chaque ligne

pour y de H à 0 :

pour x de W à 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) $f(x, y) = f(x+a, y+b) + \Delta f^{(a,b)}(x, y)$

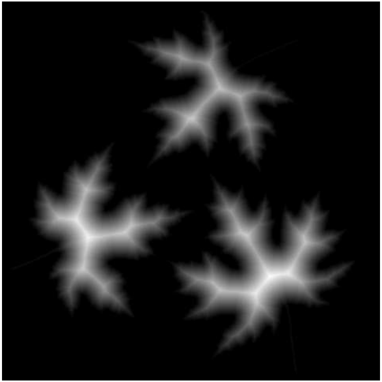
pour x de 0 à W :

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

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

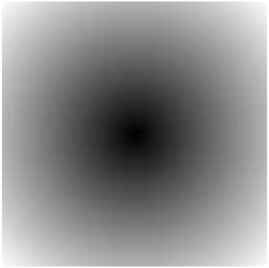
$f(x, y) = f(x-1, y) + \Delta f^{(-1,0)}(x, y)$

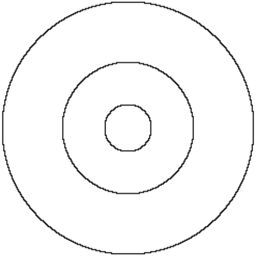
Fonction distance euclidienne : illustration



Fonction distance euclidienne calculée avec l'algorithme DL (après calcul de la racine carrée).

Fonction distance euclidienne : illustration





Fonction distance du complémentaire du pixel central, et cercles discrets (rayons 25, 75 et 125), obtenus avec l'algorithme DL.

Outline

Erosion et dilatation

L'*érosion* et la *dilatation* sont les opérateurs fondamentaux du traitement morphologique des images. Soit $X \subset \mathbb{Z}^n$ une image binaire ; $B \subset \mathbb{Z}^n$ un *élément structurant*.

Erosion morphologique

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

Dilatation morphologique

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

Erosion et dilatation

Lorsque l'élément structurant B est une boule de la distance d : $B_\lambda(x) = \{y \in \mathbb{Z}^n ; d(x,y) \leq \lambda\}$, l'érosion et la dilatation se calculent par *seuillage* de la fonction distance :

Erosion par une boule

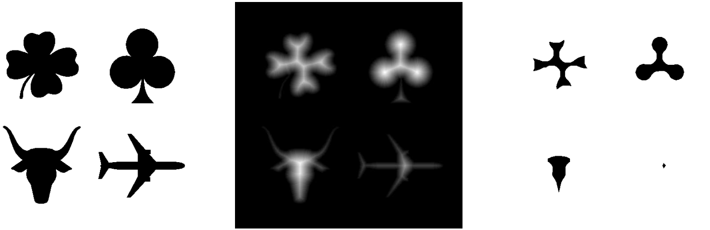
$$\varepsilon_{B_\lambda}(X) = \{z \in \mathbb{Z}^n ; d(z, X^c) \geq \lambda\}$$

Dilatation par une boule

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

Erosion morphologique

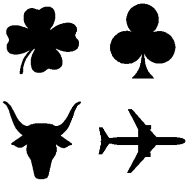
Erosion par une boule euclidienne de rayon $\lambda = 20$:



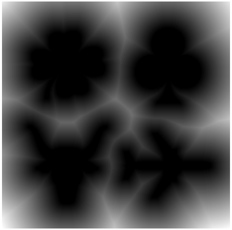
X $\quad$ F_E^X $\quad$ $\{z; F_E^X(z) \geq 20\}$

Dilatation morphologique

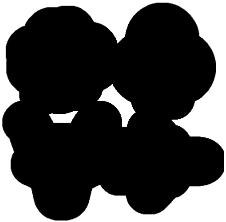
Dilatation par une boule euclidienne de rayon $\lambda = 50$:



X



$F_E^{X^c}$



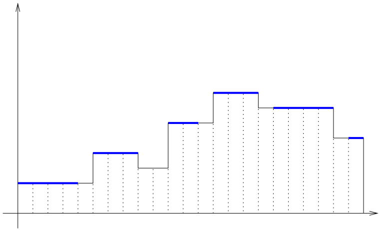
$\{z; F_E^{X^c}(z) < 50\}$

Maxima locaux et maxima régionaux

Maxima locaux

Les *maxima locaux* d'une image numérique F sont les pixels p dont la valeur $F(p)$ est plus grande ou égale à celles de ses voisins.

Maxima locaux d'une fonction numérique 1d (en bleu).

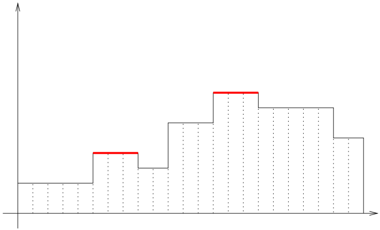


Maxima locaux et maxima régionaux

Maxima régionaux

Les *maxima régionaux* d'une image numérique F sont les plateaux (régions connexes de valeur constante), au bord desquels la valeur *diminue strictement.*

Maxima régionaux d'une fonction numérique 1d (en rouge).



Calcul des maxima locaux

Les maxima locaux se calculent *de manière locale*. Ils sont associés à une topologie : 4-connexe ou 8-connexe :

Maxima locaux 4-connexe

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

Maxima locaux 8-connexe

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

Calcul des maxima régionaux 1/2

Les maxima régionaux se calculent *de manière non locale*. Par exemple, voici l'algorithme de calcul des maxima régionaux 4-connexe.

(1) Initialisation d'une liste : maxima stricts
(i.e bords des maxima régionaux)

pour y de 0 à H :
  pour x de 0 à W :
    Si $\forall (a,b), |a|+|b|=1, F(x,y) > F(x+a,y+b)$:
      $M_4 = M_4 \cup \{(x,y)\}$
      $L = L \cup \{(x,y)\}$

Calcul des maxima régionaux 2/2

(2) Parcours de la liste : propagation des maxima

pour tout $(x, y) \in L$:

pour tout $(a, b), |a| + |b| = 1, F(x, y) = F(x + a, y + b)$:

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

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

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

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

A la fin, M_4 contient les maxima régionaux de F, au sens de la 4-connexité.

Maxima locaux et squelette morphologique

Les maxima locaux de la fonction distance coïncident avec l'ensemble des centres des boules maximales, dit *squelette morphologique*

Squelette morphologique

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

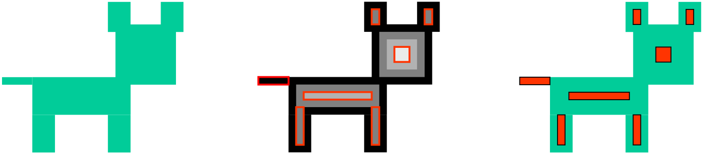
Propriété

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

Maxima locaux et squelette morphologique

Propriété

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

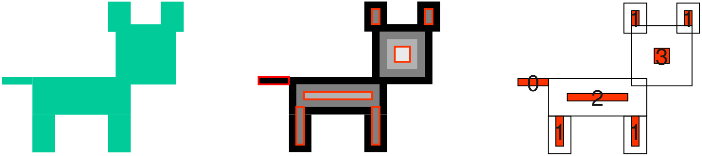


X F_8^X $S_{d_8}(X)$

Maxima locaux et squelette morphologique

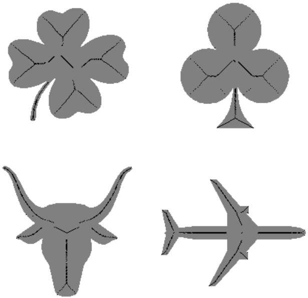
Application : Codage d'une image binaire. La donnée de la fonction distance sur le squelette morphologique fournit une représentation compacte de l'image binaire :

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

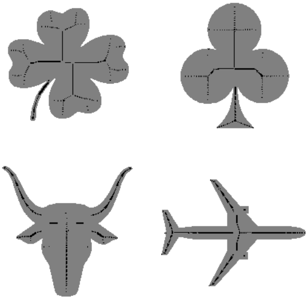


Squelette morphologique : illustrations

Squelettes morphologiques : en noir (Image originale X en gris).



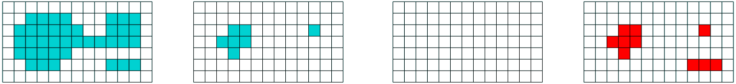
$S_{d_4}(X)$



$S_{d_8}(X)$

Maxima régionaux et érodés ultimes

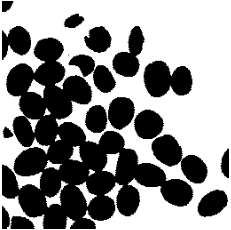
Les maxima régionaux de la fonction distance d coïncident avec un ensemble dit *érodés ultimes*, correspondant à la réunion des composantes connexes qui disparaissent entièrement lors d'une succession d'érosion réitérées par la boule de rayon 1 de la distance d.



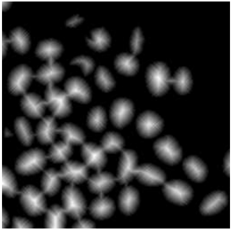
$X_0 = X$ $\quad$ $X_1 = \varepsilon_{B_8^1}(X_0)$ $\quad$ $X_2 = \varepsilon_{B_8^1}(X_1)$ $\quad$ $M_8(X)$

Maxima régionaux et érodés ultimes

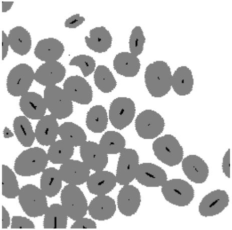
Application : Les érodés ultimes permettent de singulariser des particules qui se recouvrent mutuellement. *Exemple : comptage de cellules en imagerie quantitative*



X



F_8^X



$M_8(F_8^X)$ (en noir)

Squelettes euclidiens multi-échelles

Une des plus belles applications des fonctions distances est celle du *squelette euclidien multi-échelles.*. Le principe est le suivant :

1. Associer une *étiquette* unique à chaque pixel de contour.
2. *Propager* la valeur des étiquettes aux pixels les plus proches.
3. Calculer une *fonction de choc* locale selon la différence des étiquettes entre pixels adjacents.
4. Le *squelette* est obtenu par seuillage de la fonction de choc.

Etiquetage de contours

Soit X une image binaire.

Contour en 4-connexité

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

Contour en 8-connexité

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

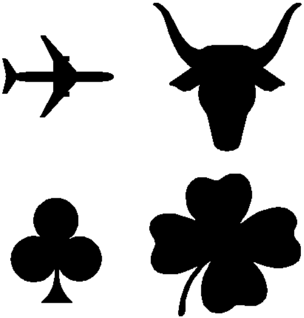
Remarque : Le contour en 4-connexité forme une courbe fermée 8-connexe pour chaque composante connexe de X. Le contour en 8-connexité forme une courbe fermée 4-connexe pour chaque composante connexe de X.

Etiquetage de contours

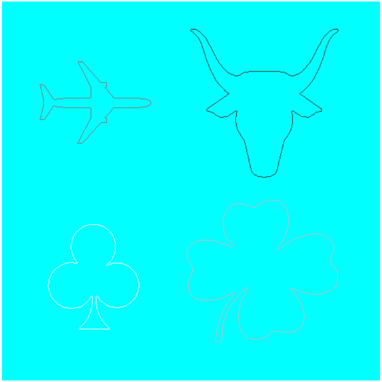
Soit X une image binaire. Soit ∂_X le contour de X. L'étiquetage de contour de X consiste à attribuer un couple d'étiquettes (Λ, λ) à chaque pixel de ∂_X, telles que :

- Λ identifie les composantes connexes de ∂_X.
- λ attribue un numéro distinct à chaque pixel de chaque composante, selon un parcours donné (e.g. sens trigonométrique).

Etiquetage de contours : Λ

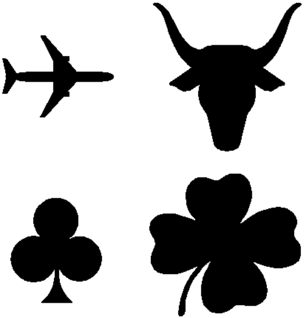


X

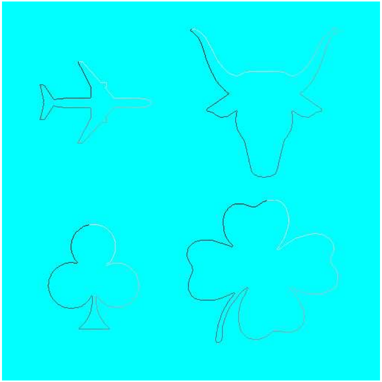


Λ_X

Etiquetage de contours : λ



X



λ_X

Etiquetage de contours 1/2

Initialisation

pour y de 0 à H :

pour x de 0 à W :

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

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

Indice $= 0$

Etiquetage de contours 2/2

Parcours des contours connexes

pour y de 0 à H :

pour x de 0 à W :

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

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

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

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

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

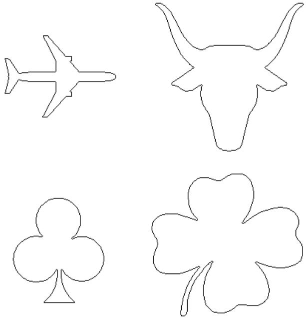
$\lambda(x', y')$ = Numero ; $(x, y) = (x', y')$;

Propagation des étiquettes

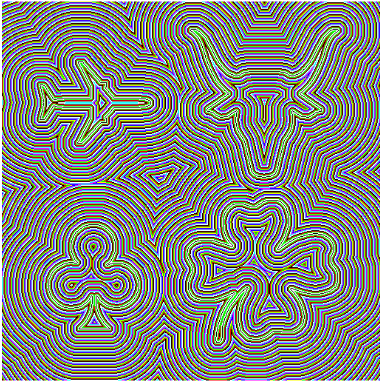
La propagation des étiquettes aux pixels les plus proches se fait tout simplement en utilisant l'algorithme de calcul de la fonction distance d sur le complementaire du contour $(\partial_X)^c$, en associant à chaque pixel (x, y) les coordonnées relatives $(R_x(x, y), R_y(x, y))$ du pixel de ∂_X le plus proche de (x, y).
Si L est une fonction étiquette sur ∂_X, la propagation de l'étiquette L selon la distance d est la fonction définie sur X comme suit :

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

Propagation des étiquettes : Illustration



$\partial^4 X$

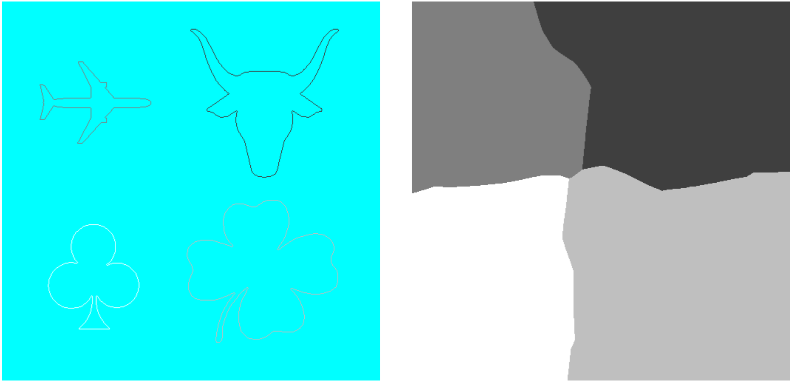


$F_E^{(\partial x)^c}$

Propagation des étiquettes : SKIZ et squelettes

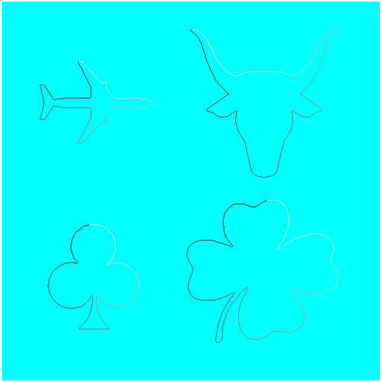
- La propagation de l'étiquette Λ (composante connexe) fournit la partition de X en zones d'influences (ou SKIZ).
- La propagation de l'étiquette λ (numérotation de contour) calcule les zones d'influences de chaque pixel de ∂_X, ce qui par différenciation, fournira le squelette de X.

Propagation des étiquettes Λ : SKIZ



Λ_X $\qquad$ $\Pi_E^{\Lambda_X}$

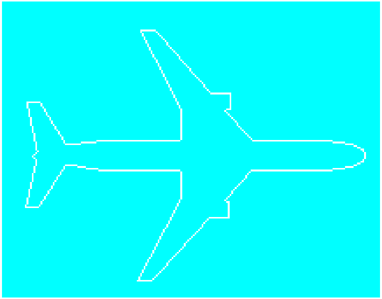
Propagation des étiquettes λ



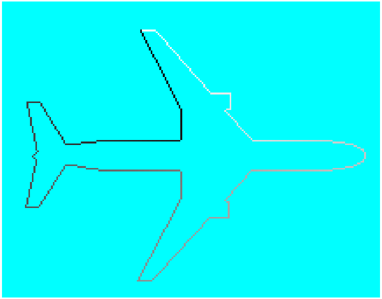
λ_X

$\Pi_E^{\lambda_X}$

Propagation des étiquettes : détail

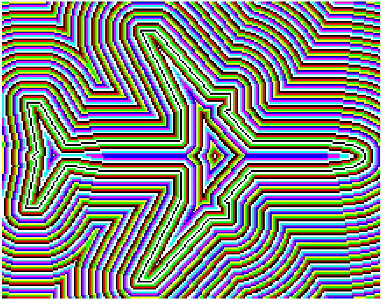


Λ_X (1 label)



λ_X (679 labels)

Propagation des étiquettes : détail



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

$\Pi_E^{\lambda x}$

Fonction de choc

- La fonction de choc associe à chaque pixel p une valeur proportionnelle à «l'éloignement» maximal entre le pixel de contour correspondant à l'étiquette de p et ceux qui correspondent à l'étiquette des pixels adjacents à p.
- L'éloignement est associé à une fonction de coût κ sur $\partial_X \times \partial_X$, où chaque pixel du contour ∂_X est identifié par son étiquette couple d'étiquette (Λ_X, λ_X).

On note $\mathcal{N}_X(p) = (p + R_x(p), p + R_y(p))$, i.e. le point de ∂_X le plus proche de p.

Fonction de choc

Fonction de choc 8-connexe

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

Fonction de choc 4-connexe

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

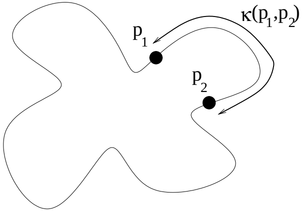
Noter la dualité : on calcule la valeur maximale dans le 4-voisinage pour un squelette 8-connexe, et réciproquement.

Fonction de choc

La fonction de coût κ associée est la *distance géodésique* entre les deux pixels $p_1 = \mathcal{N}_X(p)$ et $p_2 = \mathcal{N}_X(q)$ le long du contour ∂_X :

Fonction de coût «distance géodésique»

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

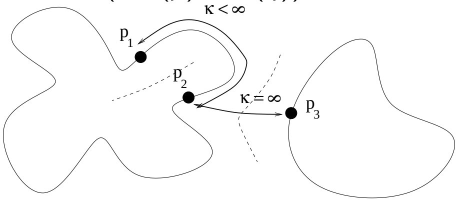


Fonction de choc

La fonction de choc se calcule très simplement en comparant les étiquettes Λ_X et λ_X des pixels voisins :

(1) Si $\Lambda_X(p) \neq \Lambda_X(q)$, alors p est à la frontière d'une zone d'influence d'un contour connexe, et :

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



Fonction de choc

(2) Si $\Lambda_X(p) = \Lambda_X(q)$ alors la fonction de coût est donnée par la différence des étiquettes λ_X, modulo le nombre de pixels total du contour :

Fonction de coût symétrique

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

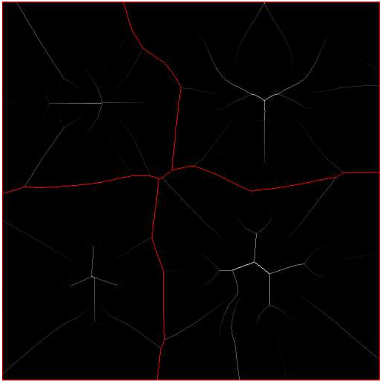
Fonction de coût asymétrique

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

La fonction de coût symétrique produit un squelette centré mais d'épaisseur 2, la fonction de coût asymétrique produit un squelette d'épaisseur 1, avec un décalage éventuel d'un demi-pixel.

Fonction de choc

$\Pi_E^{\lambda x}$



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

Squelette multi-échelle

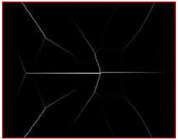
Une fois la fonction de choc S définie, le squelette à l'échelle σ est simplement défini comme le seuil de la fonction S à la valeur σ :

Squelette à l'échelle σ

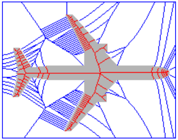
$$Sk_\sigma(X) = \{z; S(z) \geq \sigma\}$$

Connexité des squelettes multi-échelles

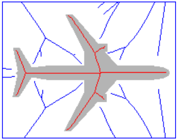
Propriété : La fonction de choc associée à la distance géodésique le long du contour est *connexe-monotone*, c'est-à-dire que pour tout entier n, l'ensemble des pixels dont la fonction de choc est supérieure à n forme le même nombre de composantes connexes que l'image X initiale.



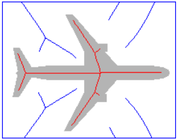
S_8



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



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

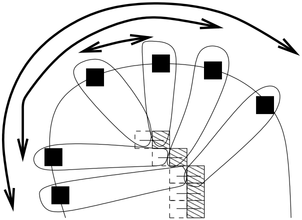


$\{z; S_8(z) > 20\}$

Connexité des squelettes multi-échelles

La propriété de connexe-monotonie de la fonction de choc est liée au fait que les zones d'influences des pixels du contours sont *connexes*.

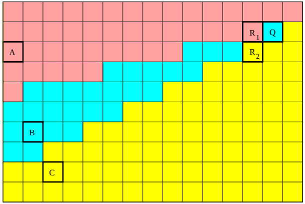
La connexité des zones d'influence des pixels implique la croissance de la fonction de choc le long des courbes du squelette à partir des extrémités :



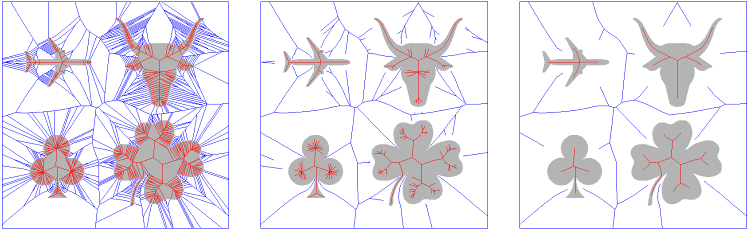
Connexité des squelettes multi-échelles

La connexité des zones d'influences est aussi une condition nécessaire de connexité des squelettes multi-échelles. L'algorithme DL est donc *plus approprié* qu'une vraie distance euclidienne pour le calcul des squelettes connexes.

Si l'on se base sur les vraies distances euclidiennes, on peut construire un chemin connexe formé de A, B et C, qui génèrera un squelette déconnecté :



Squelette multi-échelle



$Sk_1(X)$ $Sk_5(X)$ $Sk_{50}(X)$

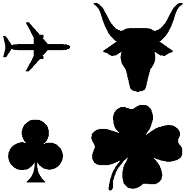
Reconstruction multi-échelle

La reconstruction à l'échelle σ est obtenue par la formule d'inversion du squelette :

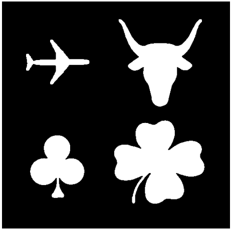
Reconstruction à l'échelle σ

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

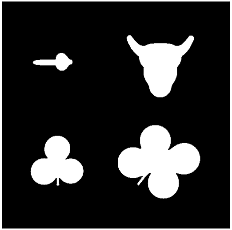
Reconstruction multi-échelle



Echelle 1



Echelle 20



Echelle 100

Carte de reconstruction

L'ensemble des reconstructions multi-échelle peut être obtenu très rapidement à partir de la carte de reconstruction définie par :

Carte de reconstruction

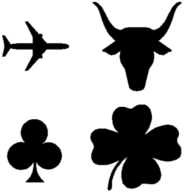
$$M_X(p) = \max_{z \in Sk_1(X); p \in B_z(F_X^E(z))} S_X(z)$$

La reconstruction à l'échelle σ se calcule alors par simple seuil de la fonction M_X :

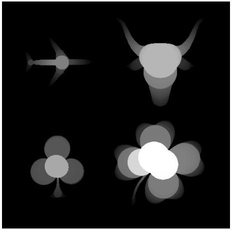
Reconstruction à l'échelle σ

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

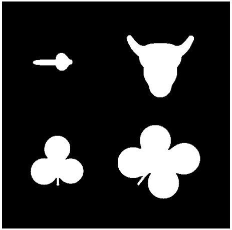
Carte de reconstruction



X



M_X

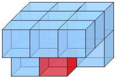


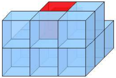
$\{z; M_X(z) > 100\}$

Extension 3d ou n-d ?

Le calcul des fonctions distance se généralise facilement aux dimensions supérieures :

1. En adaptant le voisinage de calcul :





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

2. En modifiant la fonction d'incrément $\Delta f^{\mathbf{a}}(\mathbf{x})$:

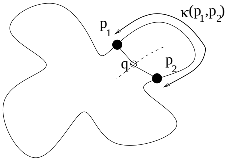
$$F_E^X(\mathbf{x}+\mathbf{a}) = F_E^X(\mathbf{x}) + \Delta f^{\mathbf{a}}(\mathbf{x}) = F_E^X(\mathbf{x}) + \sum_{i=1}^{n} 2|a_i R_{x_i}| + a_i^2$$

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

Extension 3d ou n-d ?

En revanche, la fonction de choc utilisée en 2d n'est plus utilisable en 3d :

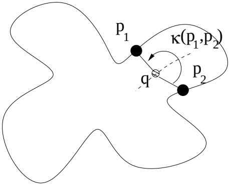
En 3d, la distance géodésique le long du contour entre 2 points du contour ne peut plus se calculer par une simple différence des étiquettes, comme en 2d : $S(q) = \kappa(p_1, p_2) = d_{\partial \mathbf{X}}(p_1, p_2)$



Extension 3d ou n-d ?

Pour cette raison, on utilisera en 3d une fonction de choc plus facilement calculable :

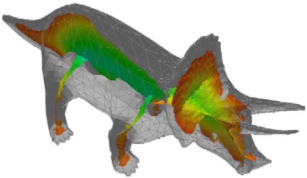
On peut définir une autre fonction de choc à partir de la mesure de l'angle entre les 2 points de contour les plus proches : $S(q) = \kappa(p_1, p_2) = \widehat{p_1 q p_2}$



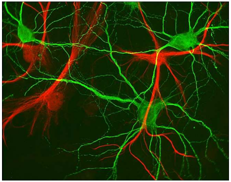
Remarque : mais la préservation de la topologie n'est plus assurée !

Applications

- Codage, compression, synthèse de formes
- Identification de modèles géométriques, topologiques.
- Morphométrie.



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



(from P. de Koninck, Univ. Laval)

Outline

1. Formalisme, Définitions et Notations
   - Pavages, Maillages et Images discrètes
   - Topologies dans la maille carrée
   - Distances dans $\mathbb{Z}^n$
2. Fonctions distances : Algorithmes
   - Algorithmes de base
   - Distances quasi-euclidiennes
3. Applications des fonctions distances
   - Opérateurs morphologiques
   - Squelette morphologique et érodés ultimes
   - Squelettes connexes multi-échelles
4. Conclusion

Conclusion

- Fonction discrètes : outil puissant pour le traitement, l'analyse et la synthèse de formes.
- Algorithmes efficaces : complexité linéaire, extension multi-dimensionnelle.
- Perspectives : pavages irréguliers, graphes quelconques.

Bibliographie

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)

Bibliographie

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)