

Segmentation et Analyse d'images (partie 2)

Anne Vialard

LaBRI, Université Bordeaux 1

Sommaire

1 Géométrie et topologie discrètes

Régions, courbes et surfaces

Représentation d'une partition de l'image

Droites et plans discrets

Distances discrètes

2 Squelettisation

3 Caractéristiques géométriques d'une frontière discrète

4 Caractéristiques géométriques d'une région

1 - Géométrie et topologie discrètes

Régions, courbes et surfaces

Représentation d'une partition de l'image

Droites et plans discrets

Distances discrètes

Connexités 2D I

- *Plan discret* : \mathbb{Z}^2
- *Point discret* : point de \mathbb{Z}^2 , correspond au pixel centré en ce point
- 2 points discrets P et Q sont dits *4-connexes* ssi

$$|x_P - x_Q| + |y_P - y_Q| = 1$$

- 2 points discrets P et Q sont dits *8-connexes* ssi

$$\max(|x_P - x_Q|, |y_P - y_Q|) = 1$$

-	+	-	-	+	-
-		-		-	

Point discret

-	+	-	○	-	+	-
-	○	-	○	-		

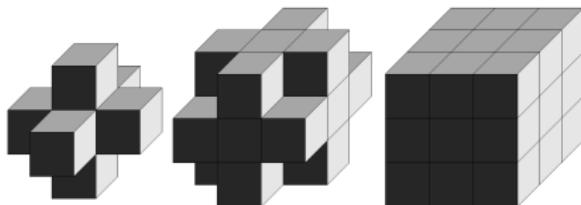
4-connexité

-	○	-	○	-
-	○	-	○	-

8-connexité

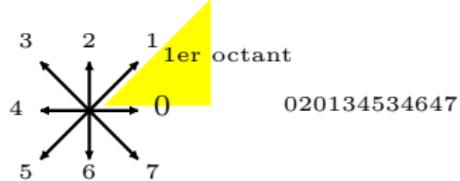
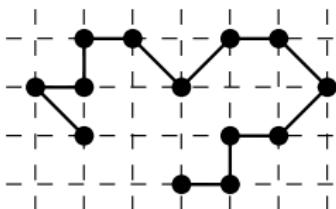
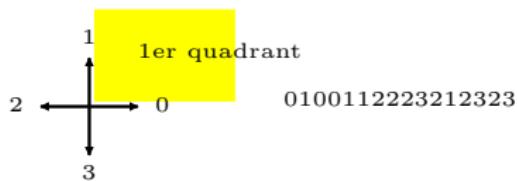
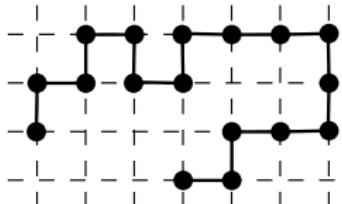
Connexités 3D I

- *Plan discret* : \mathbb{Z}^3
- *Point discret* : point de \mathbb{Z}^3 , voxel centré en ce point
- 2 points discrets P et Q sont dits **6-connexes** ssi
 $|x_P - x_Q| + |y_P - y_Q| + |z_P - z_Q| = 1$
- 2 points discrets P et Q sont dits **18-connexes** ssi
 P et Q sont 26-connexes et
 $|x_P - x_Q| + |y_P - y_Q| + |z_P - z_Q| \leq 2$
- 2 points discrets P et Q sont dits **26-connexes** ssi
 $\max(|x_P - x_Q|, |y_P - y_Q|, |z_P - z_Q|) = 1$



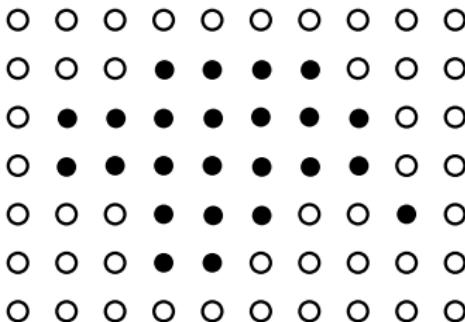
Chemin discret I

- Un *chemin discret* k -connexe est une suite de points discrets (P_0, P_1, \dots, P_n) tels que $\forall i \in 1..n$, P_{i-1} et P_i sont k -adjacents.
- *Code de Freeman* : (P_0, \dots, P_n) est représenté par $(P_0, d_0, \dots, d_{n-1})$. La direction d_i code le déplacement élémentaire de P_i à P_{i+1} .



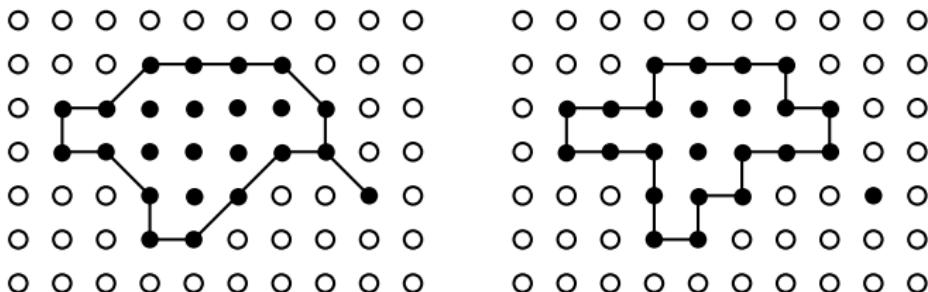
Composante connexe / région I

- **Ensemble connexe** : ensemble de points discrets E tel que $\forall P, Q \in E, \exists$ un chemin discret (M_0, \dots, M_n) tel que $M_i \in E$, $M_0 = P$, $M_n = Q$.
- **Composante connexe** d'un ensemble de points discrets : ensemble connexe maximal (ou classe d'équivalence pour la relation d'adjacence).
- Exemple : ensemble discret composé d'une composante 8-connexe (de 2 composantes 4-connexes)



Frontière 2D, définition classique I

La *frontière* d'une région 8-connexe (respectivement 4-connexe) R est l'ensemble des points de R dont au moins un des 4-voisins (resp. 8-voisins) n'est pas élément de R .
⇒ La frontière est composée de chemins 8-connexes (resp 4-connexes).

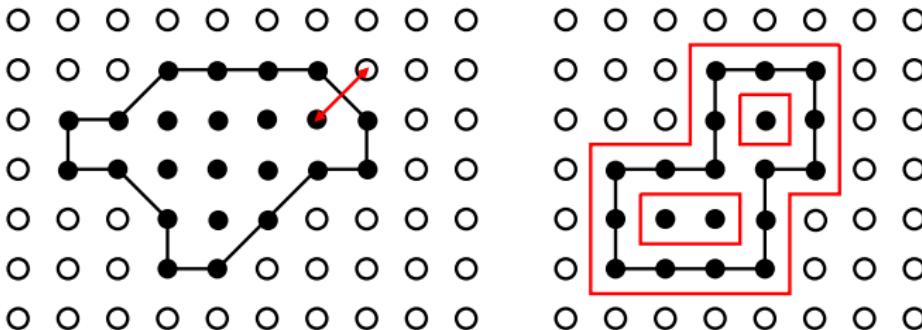


Problèmes :

- 2 régions adjacentes n'ont pas de portion de frontière commune

Frontière 2D, définition classique II

- Une frontière 8-connexe ne sépare pas le plan discret en deux composantes 8-connexes distinctes.
- Une frontière 4-connexe peut séparer le plan en plus de 2 composantes 4-connexes distinctes.



Théorème de Jordan 2D I

Une *k-courbe fermée C* est un chemin discret *k*-connexe tel que tout point de *C* est adjacent à exactement 2 autres points de *C*.

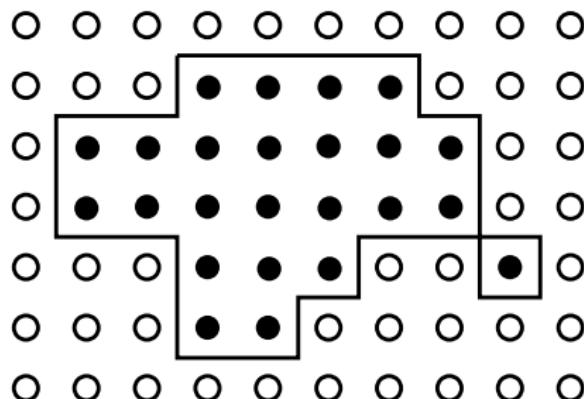
Théorème : Soient $k \in \{4, 8\}$, $\bar{k} = 12 - k$, *C* une *k*-courbe fermée de \mathbb{Z}^2 . Le complémentaire de *C* est formé d'exactement deux composantes \bar{k} -connexes.

Problème : les frontières définies précédemment ne sont pas des *k*-courbes fermées...

Frontière 2D, définition interpixels I

Chaque point discret de la région est vu comme un pixel (carré unitaire de côté 1 centré sur le point).

La frontière interpixel est une *suite d'arêtes* de pixels du bord de la région. On peut représenter cette frontière par un chemin discret 4-connexe (décalé dans la grille demi-entièrre).

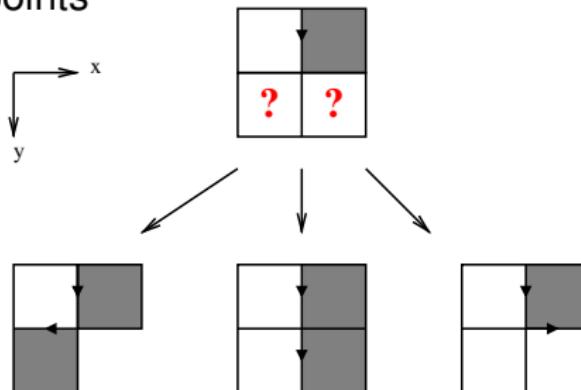


Suivi de frontière interpixel I

Objet 8-connexe

Hypothèse : la région se trouve à gauche du contour.

- Recherche du 1er point (x, y) par balayage
- Initialisation du résultat à 23, de la direction courante $d = 3$
- Recherche de la direction suivante : à chaque pas on teste un ou deux points



- Fin du suivi : retour au 1er point.

Suivi de frontière interpixel II

Algorithme

$$\Delta x = \{1, 0, -1, 0\}$$

$$\Delta y = \{0, -1, 0, 1\}$$

Si $P = \{x + \Delta x[d] + \Delta x[d - 1], y + \Delta y[d] + \Delta y[d - 1]\}$ pt objet

$$d = d - 1$$

$$(x, y) = P$$

Sinon si $P = \{x + \Delta x[d], y + \Delta y[d]\}$ pt objet

$$(x, y) = P$$

Sinon

$$d = d + 1$$

Ajouter d au résultat

Suivi du bord d'une région I

Objet 8-connexe

Recherche du 1er point (x, y) par balayage, $dir = 4$

Recherche du point suivant :

Faire

$$dir = (dir + 1) \bmod 8$$

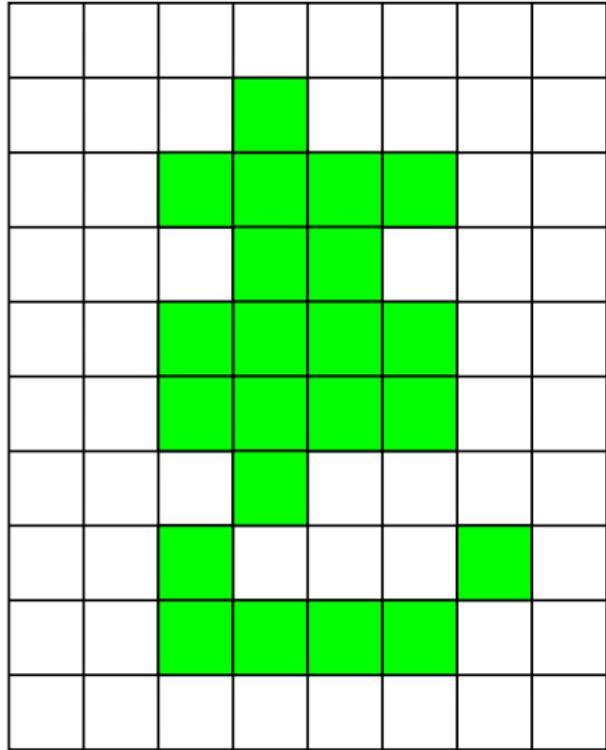
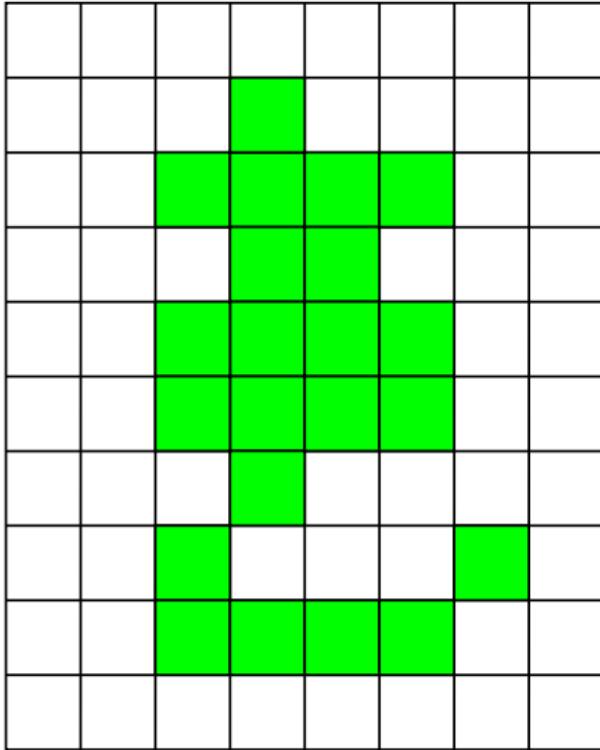
Jusqu'à ce que le point suivant (x, y) dans la direction dir soit un point de l'objet.

Ajouter dir au code de Freeman

$$dir = MAJ[dir] \quad (MAJ = \{6, 6, 0, 0, 2, 2, 4, 4\})$$

Fin du suivi : il faut retomber sur les deux 1ers points.

Suivi du bord d'une région II



Un peu de théorie I

- **Espace discret**: (V, W) où V est l'ensemble des points discrets (pixels en 2D, voxels en 3D) et W une relation d'adjacence entre points discrets (lignels, surfels).
- **Surface S** : sous-ensemble non vide de W
 - $II(S) = \{u / \exists v \in V \text{ tq } (u, v) \in S\}$
 - $IE(S) = \{v / \exists u \in V \text{ tq } (u, v) \in S\}$
 - $I(S) = \{p \in V / \exists \text{ un } W\text{-chemin de } p \text{ à } II(S) \text{ ne traversant pas } S\}$
 - $E(S) = \{p \in V / \exists \text{ un } W\text{-chemin de } p \text{ à } IE(S) \text{ ne traversant pas } S\}$
- Une surface est **presque-Jordan** ssi tout W -chemin d'un élément de $II(S)$ à un élément de $IE(S)$ traverse S .
 $\Leftrightarrow I(S) \cap E(S) = \emptyset$
- Une surface est **$\kappa\lambda$ -Jordan** ssi elle est presque-Jordan et son intérieur est κ -connexe et son extérieur est λ -connexe.

Un peu de théorie II

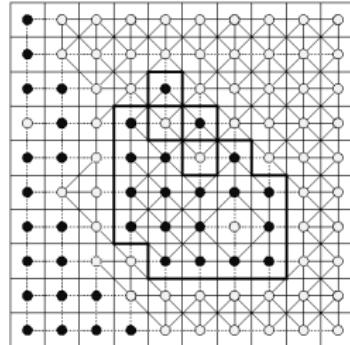
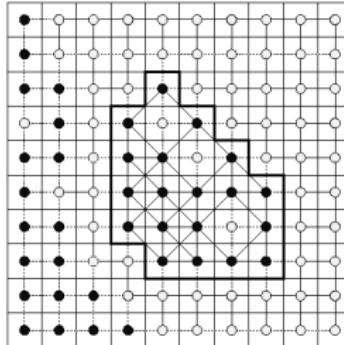
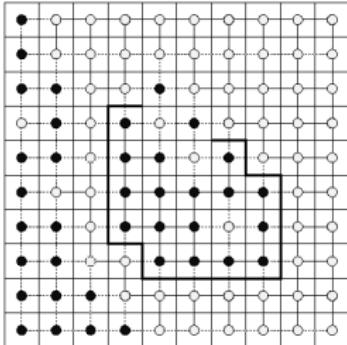
- *Frontière* de O et Q sous-ensembles de V :

$$\partial(O, Q) = \{(u, v) \in W / u \in O \text{ et } v \in Q\}.$$

Dans une image binaire :

- S est un *$\kappa\lambda$ -bord* s'il existe un objet noir O κ -connexe et un objet blanc Q λ -connexe tq $S = \partial(O, Q)$.
- *Paire de Jordan* :
 - en 2D : (8, 4), (8, 8)
 - en 3D : (18, 6), (26, 6), (14, 6)

Pour une paire de Jordan (κ, λ) , tout $\kappa\lambda$ -bord est $\kappa\lambda$ -Jordan.



Suivi de frontière 3D I

Image binaire I

B_el : surfel (u, v) tel que u noir et v blanc

Ensemble des bels de I : $B(I)$

Bel initial : b_0

Adjacence entre bels : β

Algorithme

E : ensemble des bels déjà traités

Q : file des bels touchant E et à traiter

L : liste résultat

Mettre b_0 dans Q et dans E

Tant que Q non vide

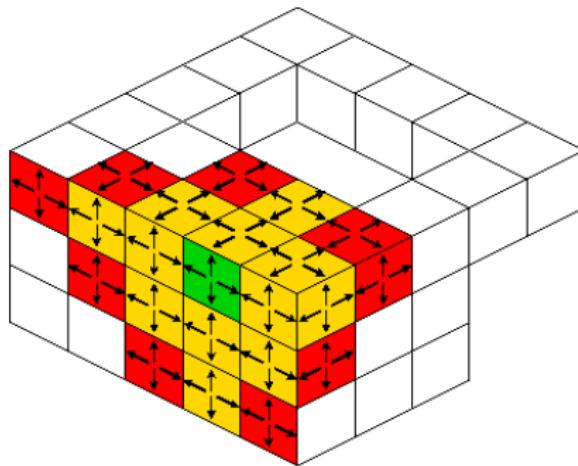
 Retirer b de Q et le mettre dans L

 Pour chaque $b' \in B(I)$ tel que $\beta(b, b')$

 Si $b' \notin E$

 Mettre b' dans E et dans Q

Suivi de frontière 3D II



1 - Géométrie et topologie discrètes

Régions, courbes et surfaces

Représentation d'une partition de l'image

Droites et plans discrets

Distances discrètes

Représentation d'une partition de l'image

Structure de données pour la segmentation

Partition de l'image \Rightarrow 2 types d'informations

- **Géométrie** : forme des régions
 - ensemble des pixels/voxels d'une région
 - région contenant un pixel/voxel donné
 - frontières, frontière d'une région
- **Topologie** : voisinages, inclusions
 - ensemble des régions voisines d'une région donnée
 - régions incluses dans une région
 - région englobante

Représentation d'une partition 2D

contours inter-pixels et carte combinatoire

- Géométrie de la partition

- *segment* : chemin maximum entre 2 régions (portion de frontière commune).
- *noeud* : intersection de plusieurs segments

L'extérieur de l'image est considéré comme une région.

- Topologie de la partition : graphe planaire avec mise en correspondance de noeud / sommet, segment / arête, région / face

Rappel : une permutation est une application bijective d'un ensemble E dans E .

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 3 & 5 & 1 & 2 & 4 \end{pmatrix} \text{ cycles : } (1, 3)(2, 5, 4)$$

Carte combinatoire 2D

La **topologie** de la partition est codée par une **carte combinatoire** : couple de permutations (σ, α)

Rq : chaque arête du graphe est décomposée en 2 brins.

La permutation σ représente **les sommets** (un sommet peut être codé par les brins rencontrés quand on tourne autour dans le sens positif).

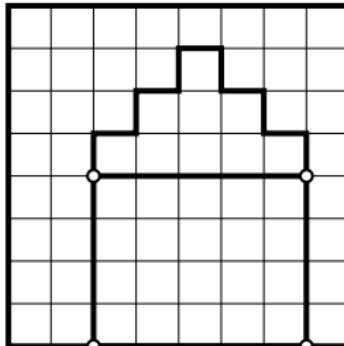
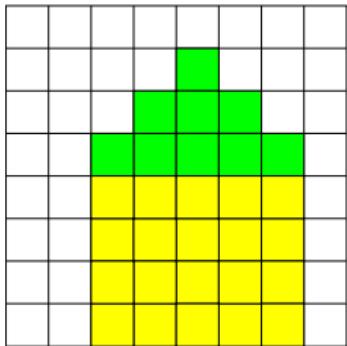
La permutation α représente **les arêtes** (= relation d'adjacence entre faces). Soit un brin b , $\alpha(b) = -b$. Chaque brin appartient à une seule face (par convention celle à sa droite).

Les cycles de la permutation $\phi = \sigma \circ \alpha$ correspondent aux **faces**. En associant une constante à chaque cycle de ϕ , on définit une fonction λ d'étiquetage de faces.

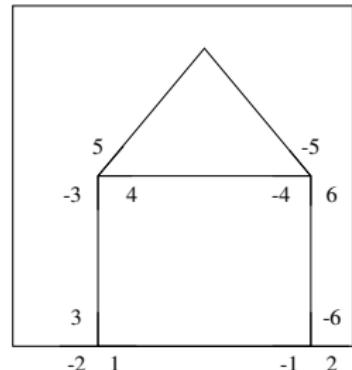
$\lambda(b) =$ face à laquelle appartient le brin b .

$\lambda^{-1}(\text{face}) =$ un brin appartenant à la face.

Exemple de carte combinatoire 2D



Géométrie



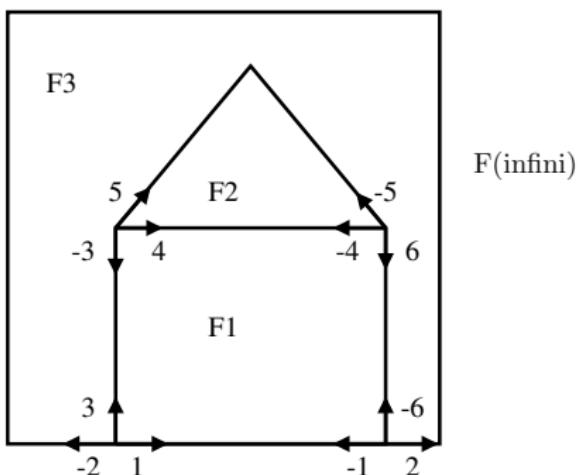
Topologie

$$\sigma = (3, -2, 1)(-6, -1, 2)(-5, -4, 6)(5, -3, 4)$$

$$\begin{matrix} \sigma = \\ \left(\begin{array}{cccccccccccc} -6 & -5 & -4 & -3 & -2 & -1 & 1 & 2 & 3 & 4 & 5 & 6 \\ -1 & -4 & 6 & 4 & 1 & 2 & 3 & -6 & -2 & 5 & -3 & -5 \end{array} \right) \end{matrix}$$

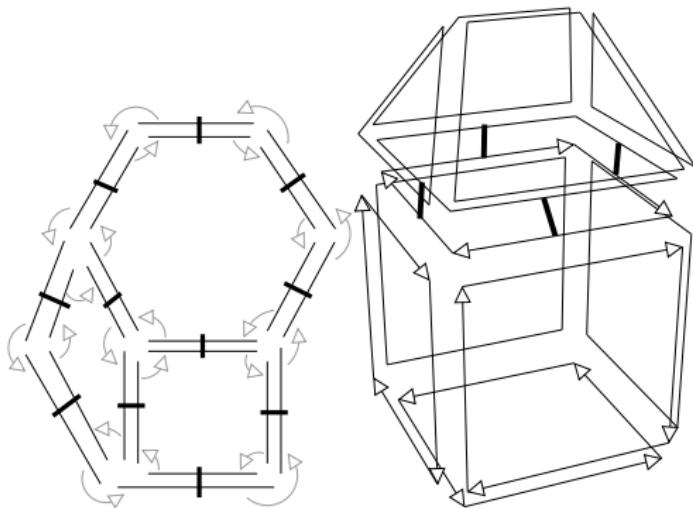
$$\alpha = (1, -1)(2, -2)(3, -3)\dots$$

Exemple de carte combinatoire 2D



$$\phi = \begin{pmatrix} -6 & -5 & -4 & -3 & -2 & -1 & 1 & 2 & 3 & 4 & 5 & 6 \\ -5 & -3 & 5 & -2 & -6 & 3 & 2 & 1 & 4 & 6 & -4 & -1 \end{pmatrix}$$
$$\phi = \underbrace{(-4, 5)}_{f_2} \underbrace{(4, 6, -1, 3)}_{f_1} \underbrace{(1, 2)}_{f_\infty} \underbrace{(-5, -3, -2, -6)}_{f_3}$$

De la dimension 2 à la dimension 3



une involution supplémentaire : relation d'adjacence entre volumes

1 - Géométrie et topologie discrètes

Régions, courbes et surfaces

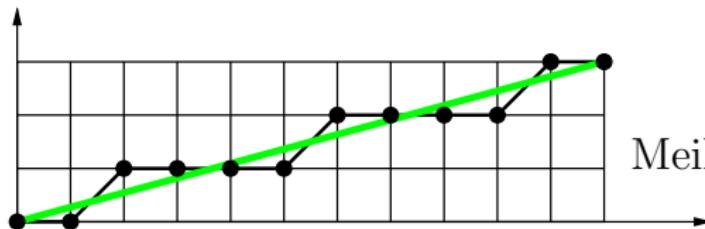
Représentation d'une partition de l'image

Droites et plans discrets

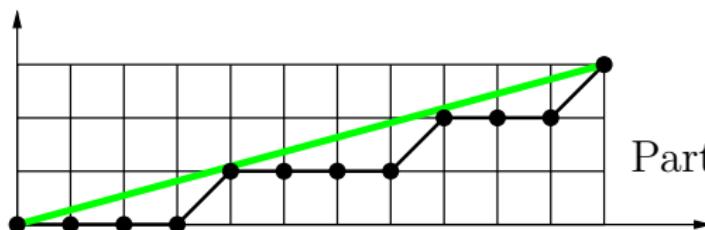
Distances discrètes

Droite discrète - Définition I

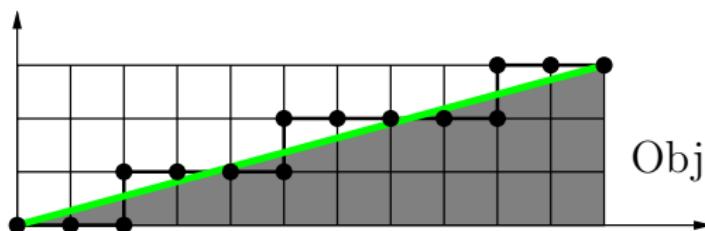
Discrétisation d'une droite réelle



Meilleur ajustement



Partie entière

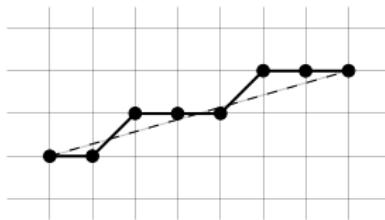


Object Boundary Quantization

Caractérisation de Rosenfeld (74) I

Un chemin discret 8-connexe C est un segment de droite discrète ssi il vérifie la **propriété de la corde** :

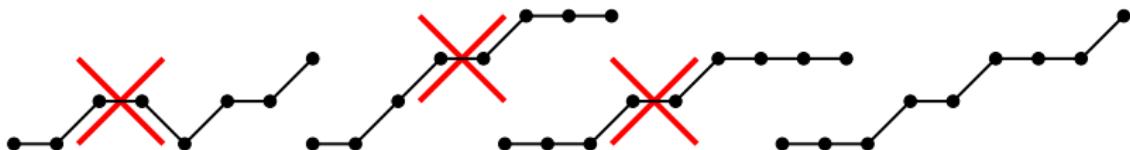
$$\forall P, Q \in C, \forall m \in [P, Q], \exists M \in C / \max(|x_M - x_m|, |y_M - y_m|) < 1$$



Caractérisation de Freeman (74) I

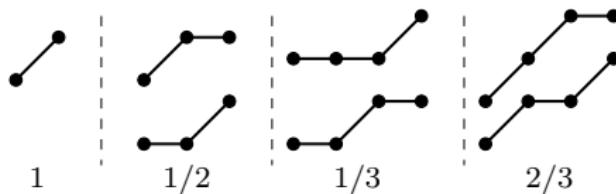
Un chemin discret 8-connexe est un segment de droite discrète ssi

- Son mot de Freeman est composé au plus de deux directions élémentaires différentes dont la différence est 1 modulo 8.
- Si le mot est composé de deux directions, une de ces directions apparaît systématiquement de manière isolée.
- Les occurrences de la direction isolée sont réparties de la manière la plus uniforme possible dans le mot.

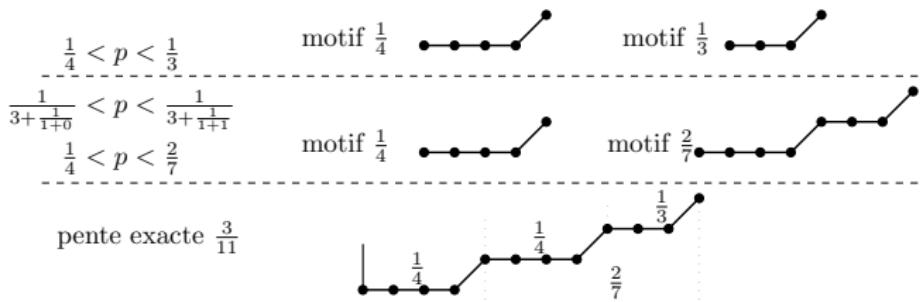


Motifs et sous-motifs I

Une droite discrète de pente rationnelle est composée d'un motif qui se répète



Exemple : droite de pente $p = \frac{3}{11} = \frac{1}{3 + \frac{1}{1 + \frac{1}{2}}}$



Définition arithmétique I

[Réveillès 91]

Caractéristiques de la droite : $(a, b, \mu) \in \mathbb{Z}^3$, $\omega \in N$

pente : $\frac{a}{b}$, position dans le plan : μ , épaisseur : ω

La droite de caractéristiques (a, b, μ, ω) est l'ensemble des points discrets vérifiant :

$$\mu \leq ax - by < \mu + \omega$$

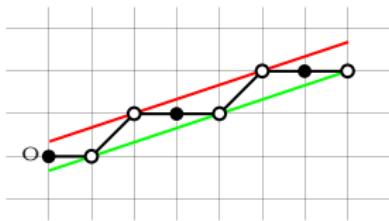
Droites d'appui : $ax - by = \mu$ et $ax - by = \mu + \omega - 1$

Points d'appui : points discrets appartenant aux droites d'appui.

$\omega = \max(|a|, |b|) \Rightarrow$ droite 8-connexe

$\omega = |a| + |b| \Rightarrow$ droite 4-connexe

Définition arithmétique II



$$(a, b, \mu) = (1, 3, -1)$$



$$(a, b, \mu) = (2, 5, -5)$$

- Droite d'appui supérieure
- Droite d'appui inférieure
- Point d'appui

Algorithme de reconnaissance I

Algorithme **incrémental** de reconnaissance de segment discret le long d'un chemin 8-connexe [Debled 95]

Principe : on ajoute un par un les points du chemin discret qu'on veut identifier à un segment de droite discrète. A chaque étape, on met à jour si nécessaire les caractéristiques du segment reconnu.

Soient $S = (P_0, \dots, P_n)$ un segment de droite discret du 1er octant et M le point à ajouter.

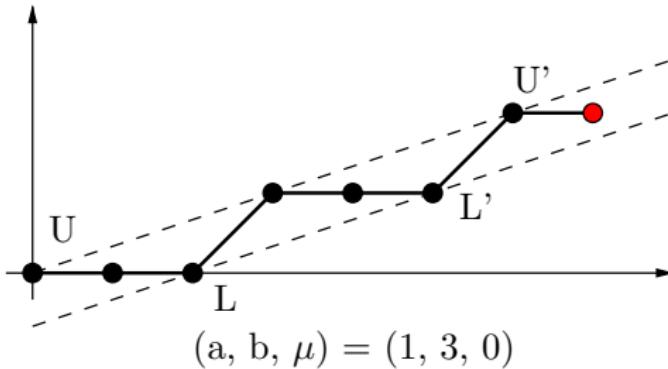
$S + M$ est-il un segment de droite ? Si oui, quelles sont ses caractéristiques ?

Condition nécessaire : M est voisin de P_n et la direction élémentaire de P_n à M est compatible avec les deux (au plus) directions composant $S \Rightarrow d(P_n, M) = 0$ ou 1

Algorithme de reconnaissance II

Cas 1 : $\mu \leq ax_M - by_M < \mu + b$

M prolonge S sans modification de caractéristiques.



U point d'appui supérieur d'abscisse minimum

U' point d'appui supérieur d'abscisse maximum

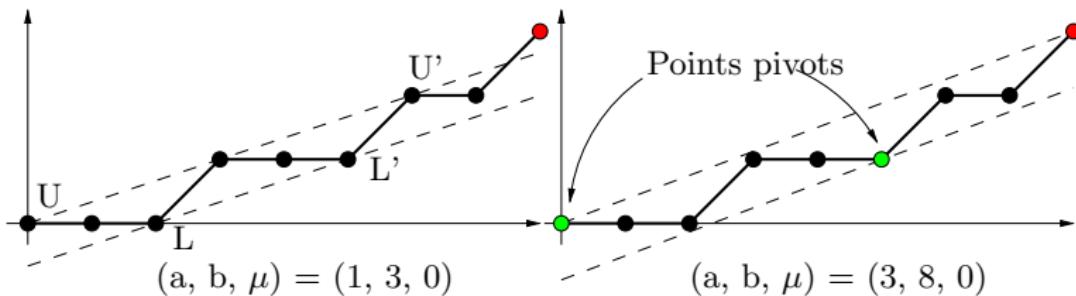
L point d'appui inférieur d'abscisse minimum

L' point d'appui inférieur d'abscisse maximum

Algorithme de reconnaissance III

Cas 2.1 : $ax_M - by_M = \mu - 1$

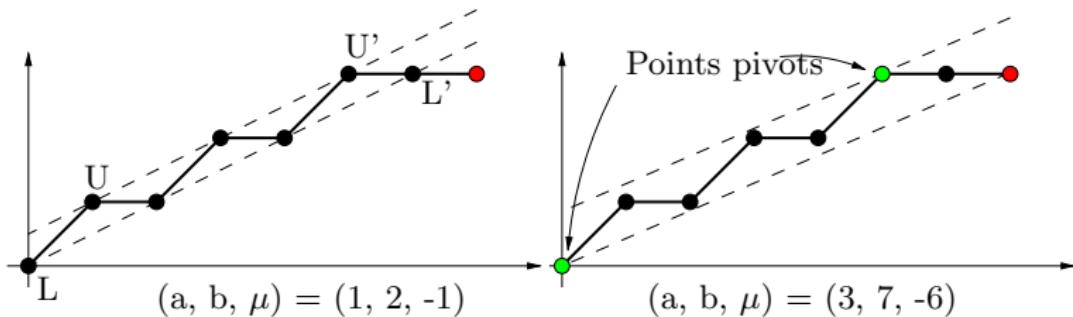
$S + M$ est un segment de caractéristiques différentes de celles de S : la pente de $S + M$ est **supérieure** à la pente de S .



Algorithme de reconnaissance IV

Cas 2.2 : $ax_M - by_M = \mu + b$

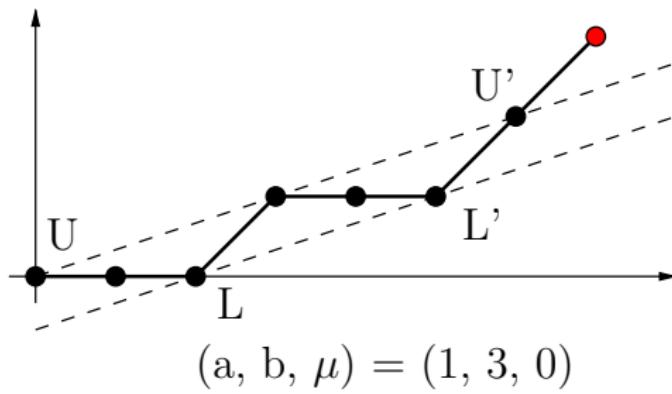
$S + M$ est un segment de caractéristiques différentes de celles de S : la pente de $S + M$ est **inférieure** à la pente de S .



Algorithme de reconnaissance V

Cas 3 : $ax_M - by_M < \mu - 1$ ou $ax_M - by_M > \mu + b$

$S + M$ n'est pas un segment de droite.



Algorithme de reconnaissance VI

Ajout d'un point

$$\text{reste} = ax_M - by_M$$

Si ($\mu \leq \text{reste} < \mu + b$)

Si ($\text{reste} == \mu$) M est point d'appui sup.

$$U' = M$$

Si ($\text{reste} == \mu + b - 1$) M est pt d'appui inf

$$L' = M$$

Sinon si ($\text{reste} == \mu - 1$) La pente augmente

$$L = L'$$

$$U' = M$$

$$a = y_M - y_U$$

$$b = x_M - x_U$$

$$\mu = ax_M - by_M$$

Algorithme de reconnaissance VII

Sinon si (*reste* == $\mu + b$) La pente diminue

$$U = U'$$

$$L' = M$$

$$a = y_M - y_L$$

$$b = x_M - x_L$$

$$\mu = ax_M - by_M - b + 1$$

Sinon M ne permet pas d'agrandir le segment

Algorithme de reconnaissance VIII

Algorithme global

(x, y) = (1, code(0))

(a, b, μ) = ($y, 1, 0$)

$U = L = (0, 0)$

$U' = L' = (1, y)$

$i = 1$

Répéter

$x = x + 1$

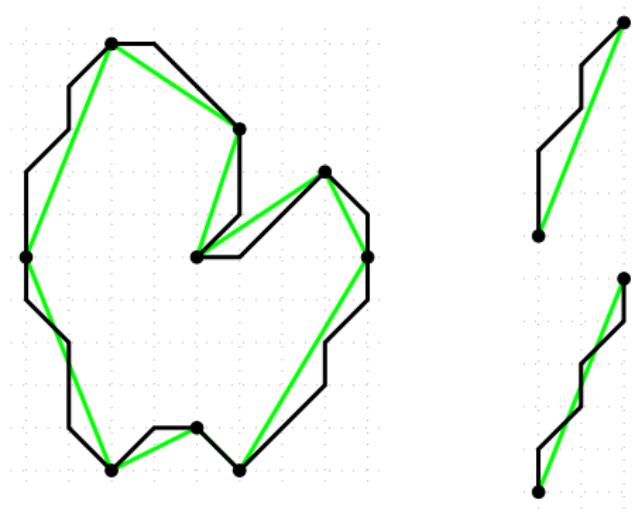
$y = y + \text{code}(i)$

$i = i + 1$

Tant que ajout-point(x, y)

Application directe : polygonalisation I

Approche itérative



Remarque : perte d'information

Autres algorithmes de polygonalisation

|

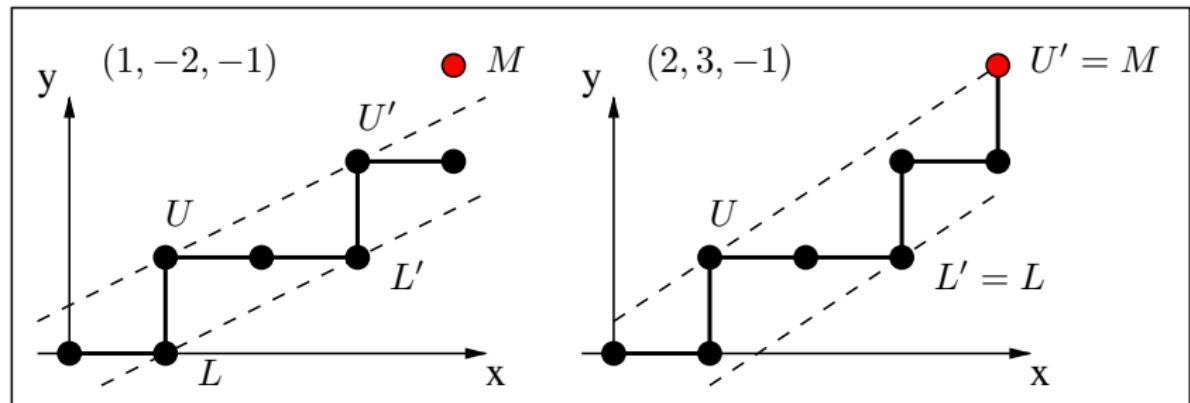
- approche itérative basé sur un calcul d'erreur (critère de distance, critère angulaire)
- approche récursive : trouver le point le plus significatif entre 2 points de contour (Douglas-Peuker, calcul de courbure)

Cas 4-connexe I

Test d'ajout d'un point M à un segment de droite :

(1) M est entre les droites d'appui : OK

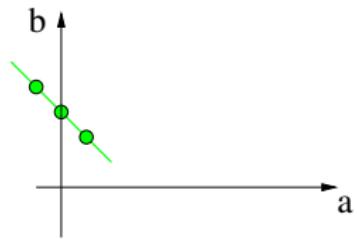
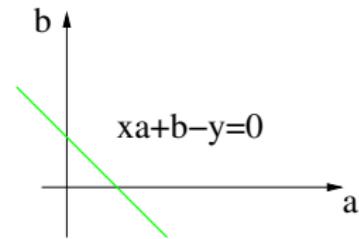
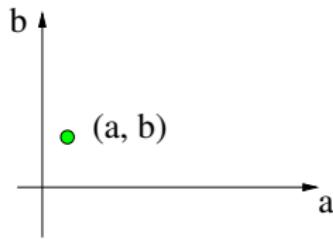
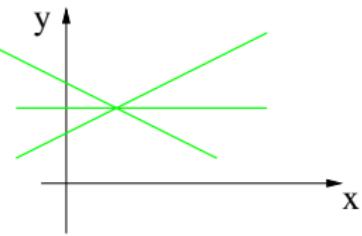
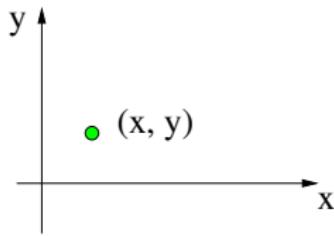
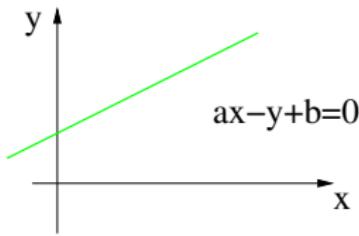
(2) $ax_M - by_M = \mu - 1$: M est "un peu trop au dessus"



(3) $ax_M - by_M = \mu + a + b$: M est "un peu trop au dessous"

Espace dual I

Autre approche pour la reconnaissance de segment de droite
discrète 2D



Espace dual II

Autre approche pour la reconnaissance de segment de droite discrète 2D

Discrétisation OBQ de $ax - y + b = 0, 0 \leq a < 1$:
ensemble des points vérifiant $0 \leq ax - y + b < 1$

Un ensemble de points (x_i, y_i) appartenant à une droite discrète est représenté dans l'espace dual par
l'intersection de bandes définies par $0 \leq x_i a + b - y_i < 1$

Reconnaitre un segment =
décider si un ensemble de contraintes linéaires est valide

Algorithme incrémental linéaire basé sur ce principe.

Plan discret I

Définition arithmétique

Le *plan de caractéristiques* $(a, b, c, \mu, \omega) \in \mathbb{Z}^5$ est l'ensemble des points discrets vérifiant :

$$0 \leq ax + by + cz + \mu < \omega$$

(a, b, c) : vecteur normal

μ : position dans l'espace

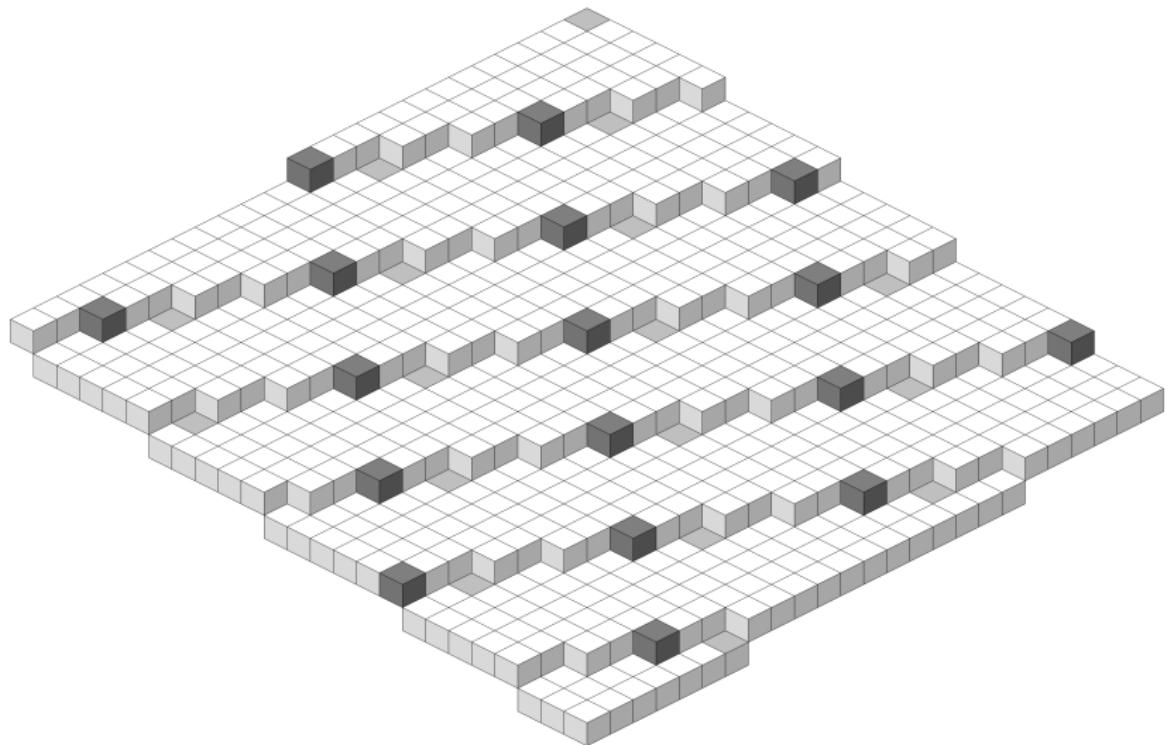
ω : épaisseur

$\omega = \max(|a|, |b|, |c|) \Rightarrow$ plan naïf (18-connexe)

$\omega = |a| + |b| + |c| \Rightarrow$ plan standard (6-connexe)

Plan discret II

Exemple : plan naïf (3, 7, 37, 0)



Plan discret : reconnaissance

[Sivignon 04]

V ensemble de voxels contenant $(0, 0, 0)$

Question : quel est l'ensemble S de paramètres

$(\alpha, \beta, \gamma), 0 \leq \alpha \leq \beta < 1, 0 \leq \gamma \leq 1$ tels que tous les voxels de V appartiennent à la discréttisation OBQ de $\alpha x + \beta y + z + \gamma = 0$?

espace des coordonnées	espace des paramètres
plan	point
point	plan
un voxel d'un plan discret	zone entre 2 plans parallèles

Construction de S : intersection de demi-espaces à chaque ajout de voxel

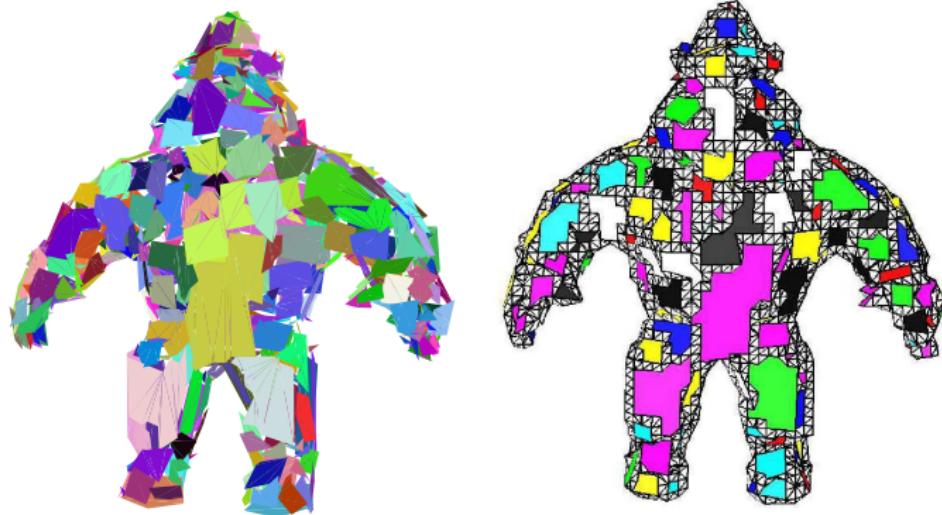
Résultat = polyèdre, polygone, segment de droite ou ensemble vide.

Polyédrisation : un exemple I

Première approche : décomposer la surface en morceaux de plans discrets + construire une courbe polygonale pour chaque bord.

Autre approche : simplifier le résultat d'un marching cube

Polyédrisation : un exemple II



1 - Géométrie et topologie discrètes

Régions, courbes et surfaces

Représentation d'une partition de l'image

Droites et plans discrets

Distances discrètes

Définition

Une *distance discrète* est une application $d : \mathbb{Z}^n \times \mathbb{Z}^n \rightarrow \mathbb{N}$ vérifiant $\forall P, Q, R \in \mathbb{Z}^n$

- $d(P, Q) \geq 0$ [positive]
- $d(P, Q) = 0 \Leftrightarrow P = Q$ [définie]
- $d(P, Q) = d(Q, P)$ [symétrique]
- $d(P, Q) \leq d(P, R) + d(R, Q)$ [triangulaire]

Deux distances classiques :

- $d_1(P, Q) = \sum_{i=0}^{n-1} |P_i - Q_i|$
- $d_\infty(P, Q) = \max_i\{|P_i - Q_i|\}$

En 2D : d_4 et d_8

En 3D : d_6 et d_{26}

très différentes de la distance euclidienne...

Distances de chanfrein

- On associe une valeur entière à chaque déplacement élémentaire dans un voisinage donné. Les déplacements élémentaires et leur valeur (coût) sont définis par un masque.
- La distance entre deux points est le coût du chemin de coût minimal joignant les deux points et formé des déplacements élémentaires autorisés.

Exemple : d_4 déplacements verticaux et horizontaux de coût 1

Masques de chanfrein 2D

	1	
1	0	1
	1	

d_4

1	1	1
1	0	1
1	1	1

d_8

4	3	4
3	0	3
4	3	4

$d_{3,4}$

	11		11	
11	7	5	7	11
	5	0	5	
11	7	5	7	11
	11		11	

$d_{5,7,11}$

Masques de chanfrein 3D

	f		f	
f	e	d	e	f
	d		d	
f	e	d	e	f
	f		f	

$z = -2 / z = 2$

f	e	d	e	f
e	c	b	c	e
d	b	a	b	d
e	c	b	c	e
f	e	d	e	f

Z=-1 / Z=1

	d		d	
d	b	a	b	d
	a	0	a	
d	b	a	b	d
	d		d	

z=0

	a	b	c	d	e	f
d_1	1					
d_{18}	1	1				
quasi-euclidienne 3×3	1	$\sqrt{2}$				
optimale 3×3	0.92644	1.34065	1.65849			
quasi-euclidienne 5×5	1	$\sqrt{2}$	$\sqrt{3}$	$\sqrt{5}$	$\sqrt{6}$	3

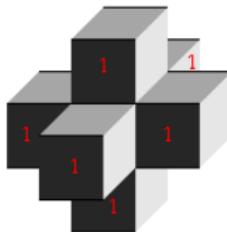
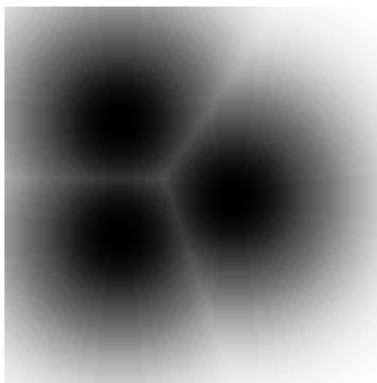


Image de distance : définition

Soit $R \subset \mathbb{Z}^n$ un ensemble de points appelés points de référence. L'*image de distance* correspondante est obtenue en associant à tout point sa distance au point de référence le plus proche.

$$\left\{ \begin{array}{l} \mathbb{Z}^n \rightarrow \mathbb{N} \\ P \rightarrow \min(d(P, P_r), P_r \in R) \end{array} \right.$$



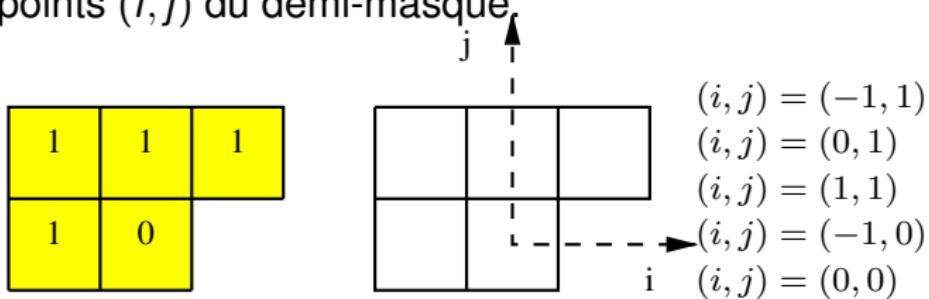
Calcul séquentiel par demi-masques I

Distances de chanfrein 2D

Principe : propagation de l'information

- Initialisation : 0 pour les pts de référence, ∞ ailleurs
- Premier balayage de l'image de gauche à droite et de haut en bas en utilisant le demi-masque avant (partie du masque décrivant la distance des points voisins se trouvant avant le centre dans le sens du balayage).

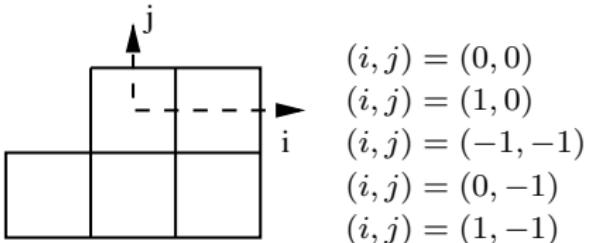
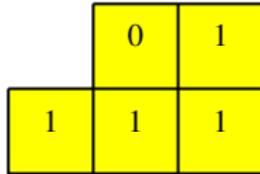
Calcul : $DM(x, y) = \min(DM(x + i, y + j) + m_{i,j})$ pour tous les points (i, j) du demi-masque



Calcul séquentiel par demi-masques II

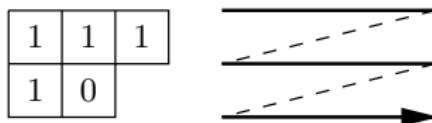
Distances de chanfrein 2D

- Deuxième balayage de l'image de droite à gauche et de bas en haut
même calcul avec le demi-masque arrière



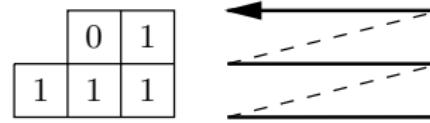
Exemple de calcul (d_8) I

Balayage avant



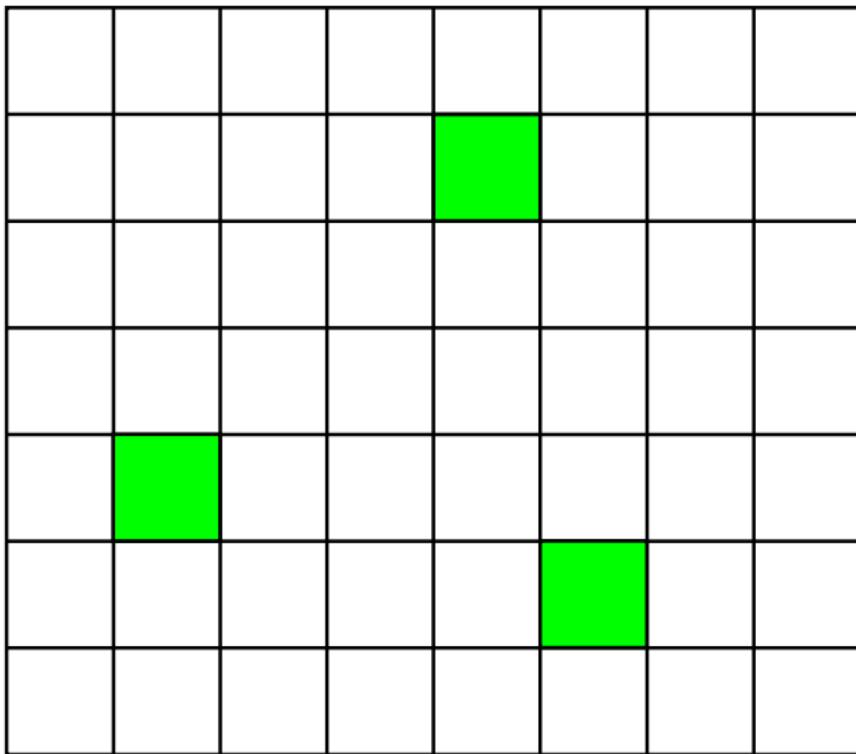
∞	∞	∞	∞	∞		1	2
∞	∞	∞	∞	1	1	1	2
∞	∞	∞	2	2	2	2	2
∞			1	2	3	3	3
1	1	1	2	3	4	4	4
2	2	2	2	3	4	5	5

Balayage arrière



3	3	3	2	1		1	2
2	2	2	2	1	1	1	2
1	1	1	2	2	2	2	2
1			1	2	3	3	3
1	1	1	2	3	4	4	4
2	2	2	2	3	4	5	5

A remplir I



Calcul séquentiel par demi-masques

Distances de chanfrein 3D

```
// Passe avant
for (z=0; z < Sz; z++)
    for (y=0; y < Sy; y++)
        for (x=0; x < Sx; x++)
            DM(x, y, z) = min∀i,j,k ∈ fp(DM(x + i, y + j, z + k) +
m[i, j, k])
```



```
// Passe arrière
for (z = Sz - 1; z ≥ 0; z--)
    for (y = Sy - 1; y ≥ 0; y--)
        for (x = Sx - 1; x ≥ 0; x--)
            DM(x, y, z) = min∀i,j,k ∈ bp(DM(x + i, y + j, z + k) +
m[i, j, k])
```

SED : Algorithme de Danielson 2D I

Question : Comment construire efficacement une image de distance euclidienne ?

- **Initialisation** : A chaque pixel sont associés 3 entiers dx , dy et $d2$. (dx, dy) donne la position du point courant relativement au point de référence le plus proche connu à un instant donné. On stocke le carré de la distance $d2 = dx^2 + dy^2$.
- **Passage descendant** : pour chaque ligne
 - parcours avant : comparaison avec points dessus / gauche
 - parcours arrière : comparaison avec point à droite
- **Passage remontant** : pour chaque ligne
 - parcours arrière : comparaison avec points dessous / droite
 - parcours avant : comparaison avec point à gauche

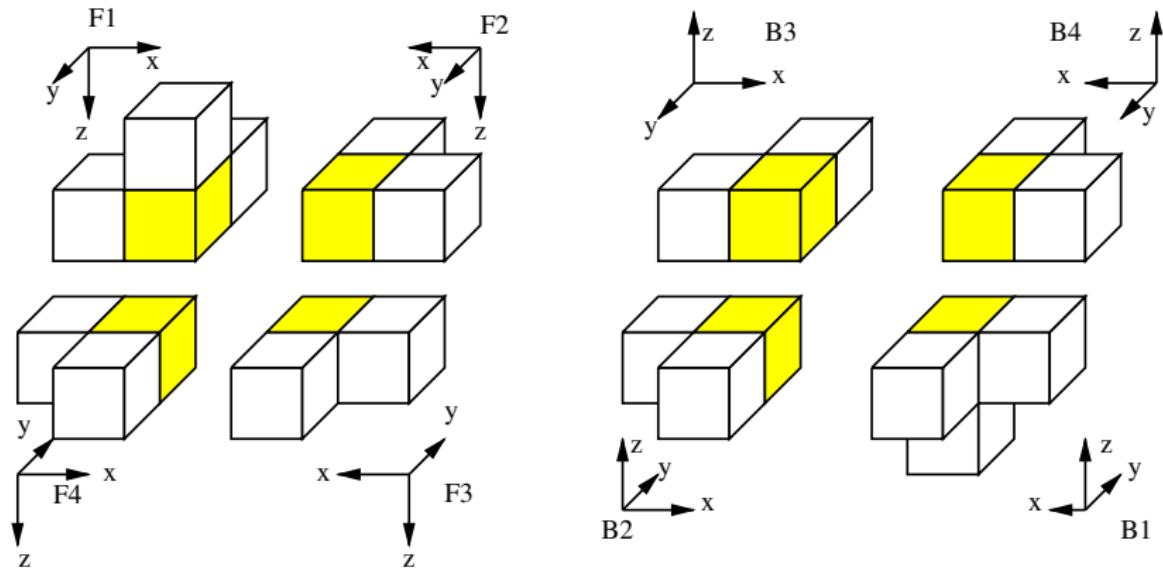
Problème : le résultat est parfois faux...

SED : Algorithme de Danielson 2D II

```
void MAJ (int x, int y, int deltaX, int deltaY)
    int d2 = DM[x+deltaX] [y+deltaY].d2
        + 2*deltaX*DM[x+deltaX] [y+deltaY].dx
        + 2*deltaY*DM[x+deltaX] [y+deltaY].dy + 1;

    if (d2 < DM[x] [y].d2)
        DM[x] [y].dx = DM[x+deltaX] [y+deltaY].dx + deltaX;
        DM[x] [y].dy = DM[x+deltaX] [y+deltaY].dy + deltaY;
        DM[x] [y].d2 = d2;
```

SED : Algorithme 3D



SED : Algorithme 3D

Détail d'une passe avant

```
for (z=0; z <  $S_z$ ; z++)
// Passe avant F1
    for (y=0; y <  $S_y$ ; y++)
        for (x=0; x <  $S_x$ ; x++)
            p=(x, y, z)
            pos = argmini||vec[p + diri] + diri||
            vec[p] = vec[p + dirpos] + dirpos
            DM(x, y, z) = ||vec[p]||
// Passe avant F2
...
// Passe avant F3
...
// Passe avant F4
...
```

Distance euclidienne exacte 2D I

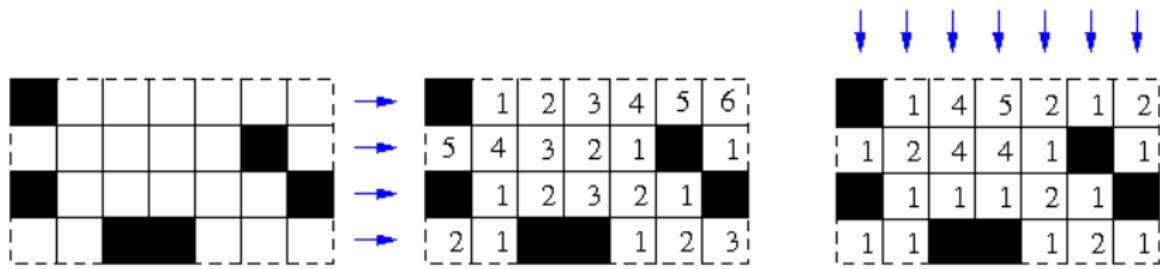
$$DT(i, j) = \min\{(i - x)^2 + (j - y)^2 : 0 \leq x < W, 0 \leq y < H, (x, y) \text{ pt de réf}\}$$

- ① Traitement des lignes : pour une ligne j

$$L(i, j) = \min_x \{|i - x| : 0 \leq x < W, (x, j) \text{ pt de référence}\}$$

- ② Traitement des colonnes

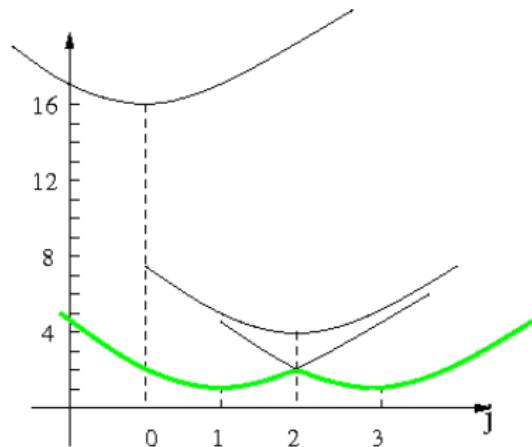
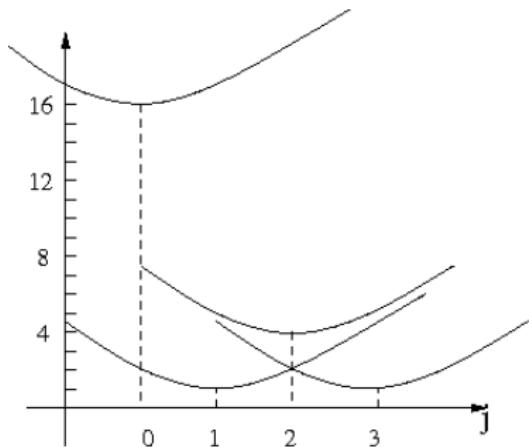
$$DT(i, j) = \min_y \{L(i, y)^2 + (j - y)^2 : 0 \leq y < H\}$$



Distance euclidienne exacte 2D II

La deuxième partie de l'algorithme peut être rendue linéaire par calcul de l'enveloppe inférieure des paraboles

$$\mathcal{P}_y^i(j) = L(i, y)^2 + (j - y)^2$$



Distance géodésique

Domaine : Objet discret R

Soit un point origine $P \in R$, on étiquette tout autre point $Q \in R$ par sa **distance géodésique** à P , longueur du plus court chemin reliant P à Q sans sortir de R .

Algorithme utilisant un masque de chanfrein de poids w :

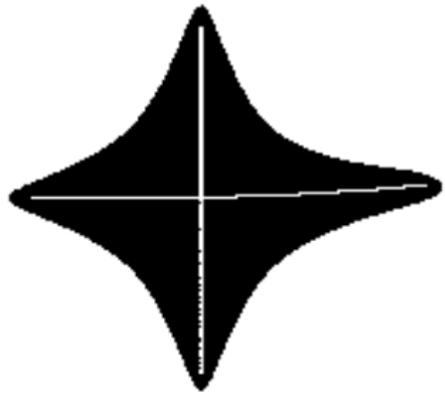
- Structure de données : ensemble de files F_i , tous les points se trouvant dans la file F_d sont à distance d du point P .
- Initialisation : Ajouter P dans F_0
- A chaque étape :
 - Dépiler Q de $F_{d_{min}}$ (file d'indice min non vide)
 - Pour chaque voisin N de Q , ajouter N dans la file $F_{d_{min}+w(\vec{QN})}$

Sommaire

- 1 Géométrie et topologie discrètes
- 2 Squelettisation
- 3 Caractéristiques géométriques d'une frontière discrète
- 4 Caractéristiques géométriques d'une région

Squelettisation

But : codage de forme, représentation simplifiée pour la reconnaissance de forme (préservation topologie, taille)



Squelette binaire

Définition (Chassery) :

Soit R une région discrète. Le squelette S_R de cette région vérifie :

- ① $S_R \subset R$
- ② S_R a le même nombre de composantes connexes et de trous que R
- ③ S_R est minimal pour la condition 2
- ④ Les régions étirées de R donnent les courbes de S_R
- ⑤ S_R est centré dans R

Remarque : une squelettisation binaire est non réversible

Calcul par amincissement

Suppression de points “inessentiels” (thinning)

- P *point simple* de $R \Leftrightarrow$

le nombre de composantes connexes de $R - \{P\} =$
le nombre de composantes connexes de R

et

le nombre de composantes connexes de $\overline{R} \cup \{P\} =$
le nombre de composantes connexes de \overline{R}

- P *point terminal* de $R \Leftrightarrow P$ a un seul voisin dans R

⇒ Idée : suppression de points simples non terminaux

Les points à considérer sont les points du bord de l'objet.

Algorithme de Thomé

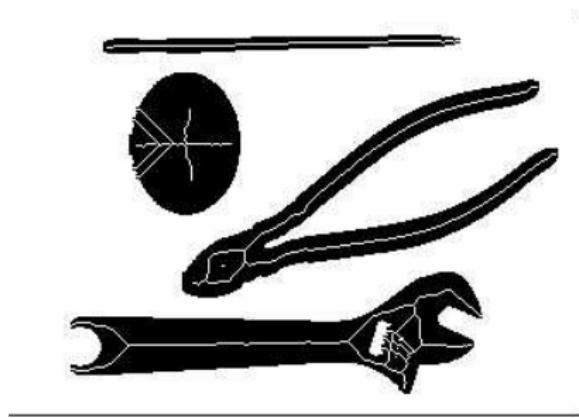
Configurations retenues pour un point situé au nord de la forme :

	0	
1	1	1
1		
1	0	0
1	1	0
0	0	
0	0	1
0	1	1
0	0	
1	0	0
1	1	0
1	1	1
0	1	1
0	0	
0	0	1
0	1	0
0	0	
1	0	0
1	1	0
1	1	1
0	0	
0	0	1
0	1	0
0	0	
1	1	0
1	1	1
0	0	
0	0	1
0	1	0
0	0	
1	0	0
1	1	0
1	1	1
0	0	
0	0	1
0	1	0
0	0	
1	1	0
1	1	1
0	0	

Algorithme :

- Traitement des points nord : on marque tous ceux qui correspondent aux configurations PUIS on supprime tous les points marqués
- Idem pour Sud, Est et Ouest
- Itérations successives jusqu'à aucun point à supprimer

Algorithme de Thomé : exemple



Images BETI :

www.tsi.enst.fr/tsi/enseignement/ressources/mti/Morpho.html

Algorithmes “d'ébarbulage”

Point simple : caractérisation locale 2D

Nombre de connexité (2D) : $T_k(P, 0) = |C_k^P[V_8^*(P) \cap O]|$

où P point discret, O ensemble de points discrets, $C_k^P(X)$ ensemble des composantes k -connexes de X k -adjacentes au point P .

$(k, \bar{k}) = (8, 4)$ connexités objet/fond

Le point P est un **point simple** de l'objet O ssi

$$T_k(P, O) = 1 \text{ et } T_{\bar{k}}(P, \bar{O}) = 1$$

Autres résultats :

- $T_k(P, O) = 0 \Leftrightarrow P$ est un point isolé
- $T_{\bar{k}}(P, \bar{O}) = 0 \Leftrightarrow P$ est un point intérieur
- $T_{\bar{k}}(P, \bar{O}) \neq 0 \Leftrightarrow P$ est un point frontière

Thinning : premier algorithme I

```
ES = ensemble des points simples de O
Tant que ES non vide
    EV = ensemble vide
    Pour chaque P de ES
        Si P est un point simple de O
            Enlever P de O
            Pour chaque Q de V(P) ∩ O
                Ajouter Q à E
    ES = ensemble vide
    Pour chaque Q de E
        Si Q est simple pour O
            Ajouter Q à ES
```

Thinnning : premier algorithme II



Thinning : algorithme directionnel

ES = ensemble des points simples de O

Tant que ES non vide

EV = ensemble vide

Pour t dans [Nord, Sud, Est, Ouest]

Pour chaque P de ES tel que type(P)=t

Si P est un point simple de O et

P n'est pas un point terminal

Enlever P de O

Pour chaque Q de V(P) \cap O

Ajouter Q à E

ES = ensemble vide

Pour chaque Q de E

Si Q est simple pour O

Ajouter Q à ES

Thinning : algorithme basé sur une fonction de priorité

Répéter

Supprimer un point P de O tel que
 P est un point simple de O et que
 $\mathcal{P}(P)$ minimal

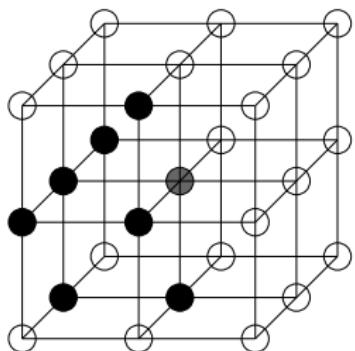
Jusqu'à aucune suppression possible

Exemple de fonction de priorité : carte de distance au fond

Thinning : extension en 3D I

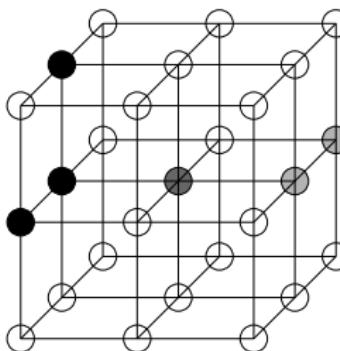
Préserver le nombre de composantes connexes de l'objet et du complémentaire + préserver les tunnels.

- *nombre de connexité 3D* : similaire au 2D, basée sur la notion de voisinage géodésique



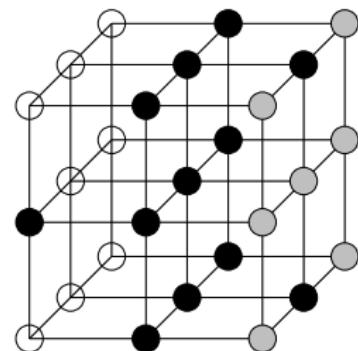
point simple

$$T_6(P, O) = T_{26}(P, \bar{O}) = 1$$



isthme 1D

$$T_6(P, O) = 2$$



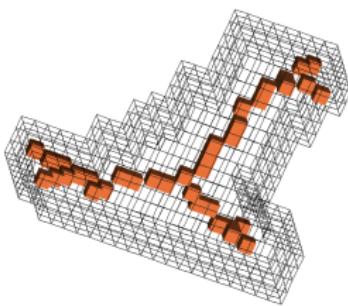
isthme 2D

$$T_{26}(P, \bar{O}) = 2$$

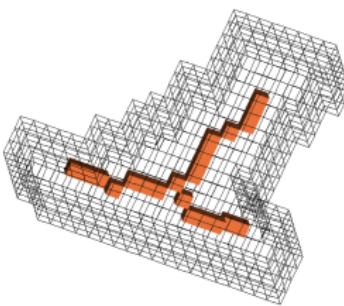
- *point simple* : idem 2D

Thinning : extension en 3D II

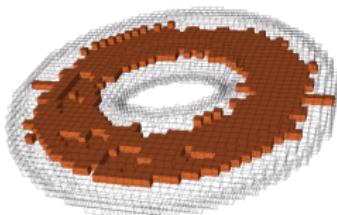
- *point terminal* : 2 types supplémentaires
 - ➊ extrémité de courbe : adjacent à un seul point de l'objet
 - ➋ isthme 1D
 - ➌ isthme 2D



(1)



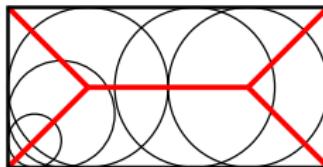
(2)



(3)

Squelettisation par axe médian

- **Boule discrète** de centre P (point discret) et de rayon r (entier) pour la distance d : $B(P, r) = \{Q / d(P, Q) \leq r\}$
- Soit R une région discrète et $P \in R$, $r_P = d(P, \overline{R}) - 1$
- L'**axe médian** de la région R est l'ensemble des boules maximales recouvrant R :
$$AM(R) = \{P \in R / \forall Q \in R, B(P, r_P) \not\subset B(Q, r_Q)\}$$



Calcul pour d_4 et d_8 :

Soit DM l'image de distance au fond. L'axe médian est composé des points correspondant aux maximums locaux de DM : $P \in AM(R) \Leftrightarrow \forall Q \in R \cap V_8(P), r_P \geq r_Q$

Sommaire

- 1 Géométrie et topologie discrètes
- 2 Squelettisation
- 3 Caractéristiques géométriques d'une frontière discrète
 - Contour discret 2D : tangente
 - Contour discret 2D : longueur / périmètre
 - Contour discret 2D : courbure
 - Extension en 3D
- 4 Caractéristiques géométriques d'une région

Géométrie d'une frontière discrète I

Problématique

Un nombre infini de formes ont la même discréétisation \Rightarrow
il n'y a pas UNE approximation unique des caractéristiques géométriques

Hypothèses sur la frontière sous-jacente : courbe lisse de courbure bornée par exemple

Estimateurs : longueur/aire, tangente/plan tangent, courbure

Propriétés souhaitées :

- convergence asymptotique
- bonne estimation à faible résolution
- préservation des propriétés de la forme (convexité par exemple)

3 - Caractéristiques géométriques d'une frontière discrète

Contour discret 2D : tangente

Contour discret 2D : longueur / périmètre

Contour discret 2D : courbure

Extension en 3D

Tangente : définition élémentaire I

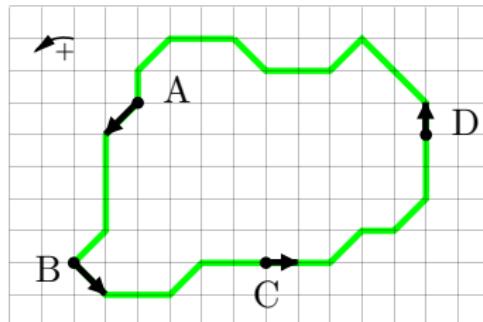
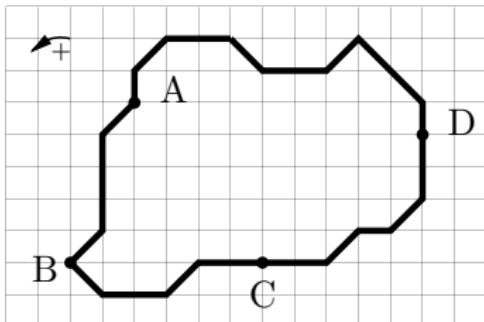
$$C = (P_0, \dots, P_n)$$

\vec{t}_i vecteur tangent au i ème point de C

$\hat{\theta}_i$ orientation de \vec{t}_i

Première approximation : $\vec{t}_i = P_i \vec{P}_{i+1}$

Nombre d'orientations différentes : 8 si contour 8-connexe



Tangente : filtrage médian I

Soit une fenêtre de taille m autour du point courant P_i .

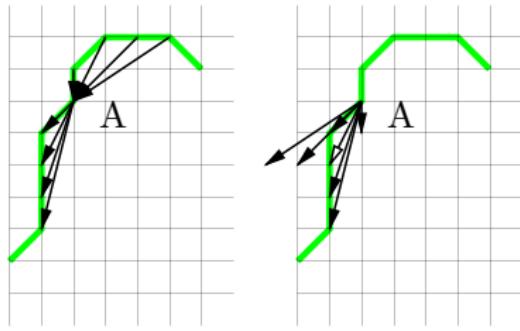
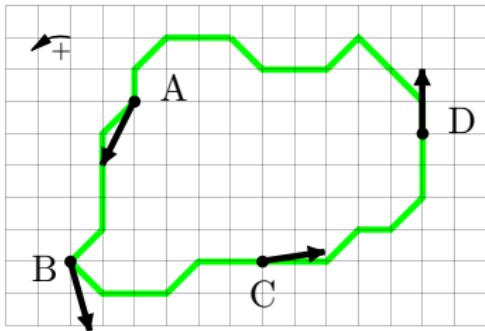
$$\vec{V}_{i,i+j} = P_i \vec{P}_{i+j} \quad j = 1..m$$

$$\vec{V}_{i,i+j} = \vec{P}_{i+j} P_i \quad j = -1..-m$$

Ces vecteurs sont classés suivant leur angle avec l'axe Ox .

Soit θ_k l'orientation du k ème vecteur dans la suite ordonnée ainsi obtenue (numérotation de 1 à $2M$).

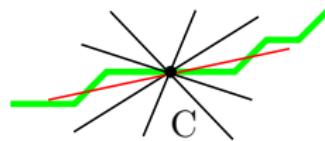
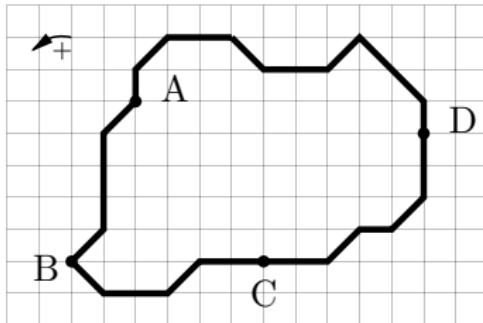
Orientation de la tangente en P_i : $\hat{\theta}_i = \frac{\theta_M + \theta_{M+1}}{2}$



Approximation de la tangente par une droite réelle I

$$\hat{\theta}_i = \operatorname{argmin}_{\Theta} \left\{ \sum_{j=-m}^m w(j) d^2(P_{i+j}, l_{\theta}) \right\}$$

$w(j)$ poids, par exemple $w(j) = G_{\sigma}(j) = \frac{1}{\sigma \sqrt{2\pi}} e^{\frac{-j^2}{2\sigma^2}}$



Approximation par une droite discrète I

Définition : *la tangente discrète symétrique* au point P_i de la courbe discrète C est la plus longue portion de C centrée en P_i qui soit un segment de droite discret.

Algorithme : ajout de paires de points autour de P_i ,
 $(P_{i-1}, P_{i+1}) \dots (P_{i-k}, P_{i+k})$ tant que $(P_{i-k}, \dots, P_{i+k})$ est un segment de droite.

⇒ adapter l'algorithme de reconnaissance d'un segment de droite discrète de façon à pouvoir étendre un segment du 1er octant/quadrant par un point d'abscisse négative.

Tangente discrète 4-connexe I

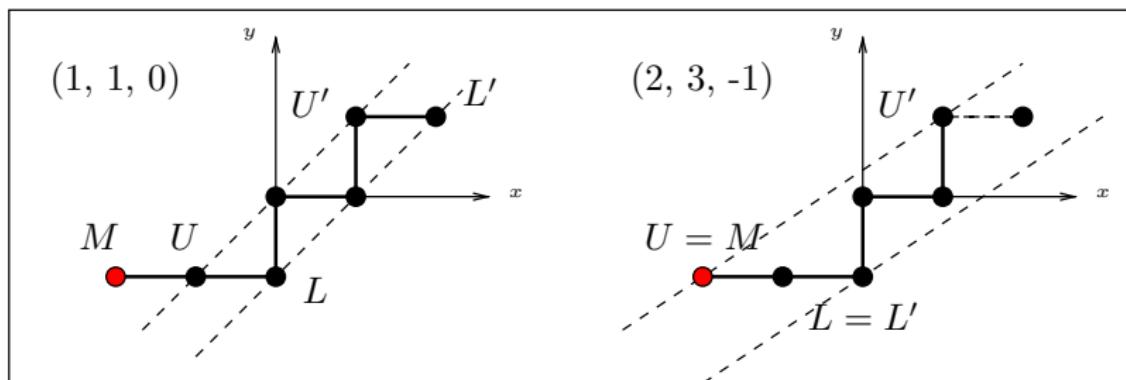
reconnaissance de segment

Ajout d'un point M d'abscisse positive : voir plus haut

Ajout d'un point M d'abscisse négative

(1) M est entre les droites d'appui : OK

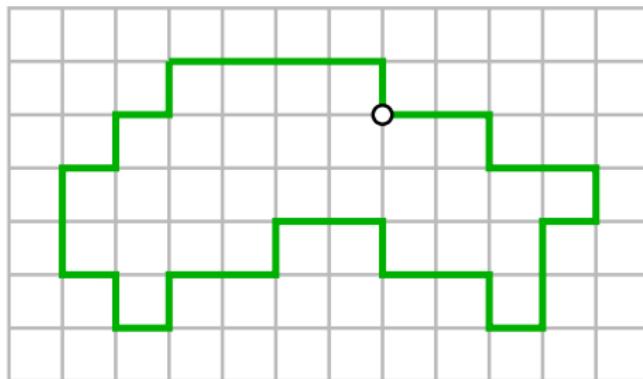
(2) $ax_M - by_M = \mu - 1$: M est "un peu trop au dessus"



(3) $ax_M - by_M = \mu + a + b$: M est "un peu trop au dessous"

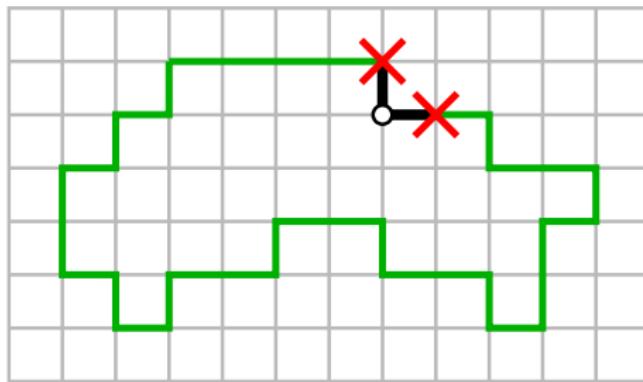
Tangente 4-connexe : exemple

Tangente symétrique autour d'un point



Tangente 4-connexe : exemple

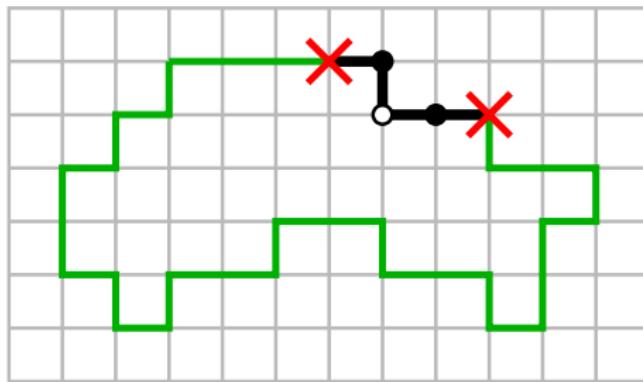
Tangente symétrique autour d'un point



$$(-1, 1, -1)$$

Tangente 4-connexe : exemple

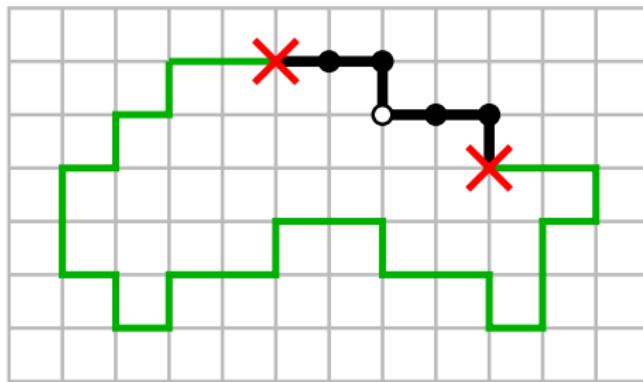
Tangente symétrique autour d'un point



$$(-1, 2, -2)$$

Tangente 4-connexe : exemple

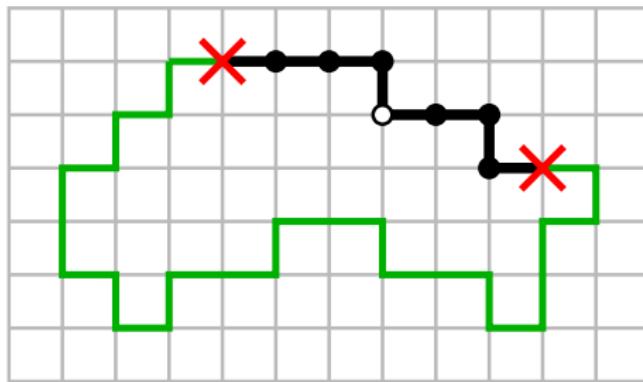
Tangente symétrique autour d'un point



$$(-1, 2, -2)$$

Tangente 4-connexe : exemple

Tangente symétrique autour d'un point



$$(-2, 5, -5)$$

Tangente discrète : bilan I

Avantages d'une définition basée sur un segment discret :

- fenêtre de taille variable suivant la configuration du contour
- algorithme indépendant du sens de parcours et du point de départ
- évaluation assez précise
- algorithme similaire en 4 ou 8-connexité

Inconvénients :

- compromis localisation/précision
- pb pour préservation de la convexité et convergence
- > autres approches (moyenne pondérée des orientations de segments de droites passant par le point considéré)

Evaluation de la tangente réelle : droite centrée entre les deux droites d'appui.

3 - Caractéristiques géométriques d'une frontière discrète

Contour discret 2D : tangente

Contour discret 2D : longueur / périmètre

Contour discret 2D : courbure

Extension en 3D

Longueur : estimateurs simples I

Les points de la courbe discrète sont répartis en k classes. On évalue alors la longueur de la courbe par :

$$\hat{L} = \sum_{i=1}^k \psi(C_i) N(C_i)$$

où $N(C_i)$ est le nombre de points de classe C_i et $\psi(C_i)$ le poids associé à cette classe.

Les poids sont estimés pour des segments de droite de pente variable \implies polygonalisation implicite.

Les estimateurs simples ne sont pas convergents

Longueur : estimateurs simples I

courbe 8-connexe

N nb de pas

N_p nb de pas horizontaux ou verticaux (direction de Freeman paire)

N_i nb de pas diagonaux (direction de Freeman impaire)

N_c nb de coins (transition entre deux directions paire et impaire)

Première évaluation de la longueur de C : $N_p + \sqrt{2}N_i$

Estimateurs :

$$\hat{L}_1 = 1.1107N$$

$$\hat{L}_K = 0.945N_p + 1.346N_i$$

$$\hat{L}_C = 0.980N_p + 1.406N_i - 0.091N_c$$

Longueur : estimateurs simples I

courbe 4-connexe

Estimateur de Rosen-Profitt :

N_c nb de coins (transition entre deux directions paire et impaire)

N_n nb de points correspondant à la jonction entre deux directions successives identiques

$$\hat{L} = \frac{\pi(\sqrt{2}+1)}{8} N_n + \frac{\pi(\sqrt{2}+2)}{16} N_c$$

Estimateur de Koplowitz :

N_{c1} nb de coins dont au moins un voisin n'est pas un coin

N_{c2} nb de coins dont les deux voisins sont des coins

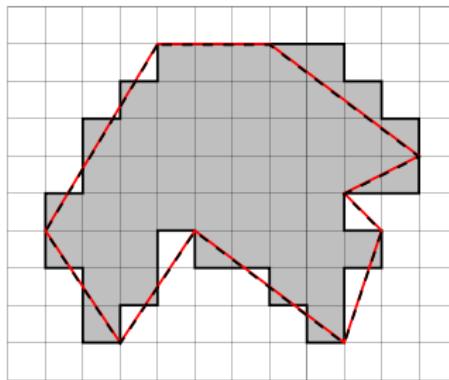
N_{n1} nb de points au centre d'une séquence rectiligne de 3 pts

N_{n2} nb de points sur une séquence rectiligne de plus de 3 pts

$$\hat{L} = 0.57736N_{c1} + 0.70251N_{c2} + 1.06681N_{n1} + 0.99350N_{n2}$$

Longueur : polygonalisation explicite I

Longueur = taille d'une approximation polygonale



Polygonalisation par droite discrète \implies estimateur de longueur convergent

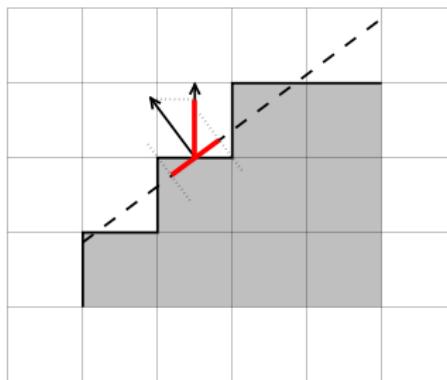
Longueur : intégration des normales I

Contribution d'une arête : $\vec{n} \cdot \vec{e}$

\vec{n} : vecteur normal calculé, \vec{e} : normale triviale

⇒ adapter le calcul de tangente pour le définir sur une arête.

Longueur totale : $\hat{L} = \sum \vec{n}_i \cdot \vec{e}_i$



Estimateur de tangente convergent ⇒ estimateur de longueur convergent

3 - Caractéristiques géométriques d'une frontière discrète

Contour discret 2D : tangente

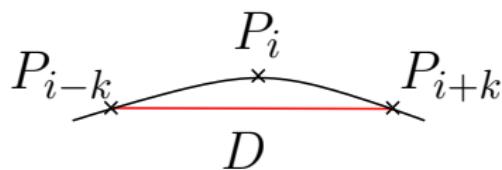
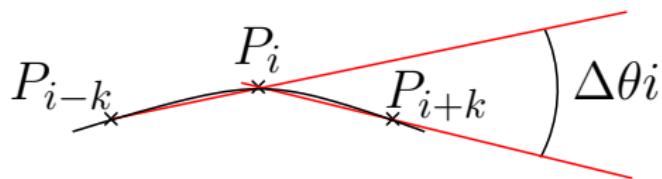
Contour discret 2D : longueur / périmètre

Contour discret 2D : courbure

Extension en 3D

Courbure : évaluation simple I

- ① Variation angulaire : $\Delta\theta_i$
- ② Rapport de distance : $\frac{d}{D}$ (d = distance le long de la courbe)



Courbure : dérivation de la tangente I

Une définition de la courbure :

dérivée de l'orientation de la tangente [Woring 93].

- Calcul par différences finies :

$$\hat{\theta}_i = \frac{\hat{\theta}_{i+k} - \hat{\theta}_{i-k}}{d(P_{i-k}, P_{i+k})}$$

ou

$$\kappa_i = \frac{\|\vec{t}_{i+k} - \vec{t}_{i-k}\|}{d(P_{i-k}, P_{i+k})}$$

- Dérivée lissée : convolution par la dérivée d'une gaussienne.

$$\kappa = \frac{\hat{\theta} * G'_\sigma}{1.1107} \quad G'_\sigma(x) = \left(\frac{-x}{\sigma^3 \sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}} \right)$$

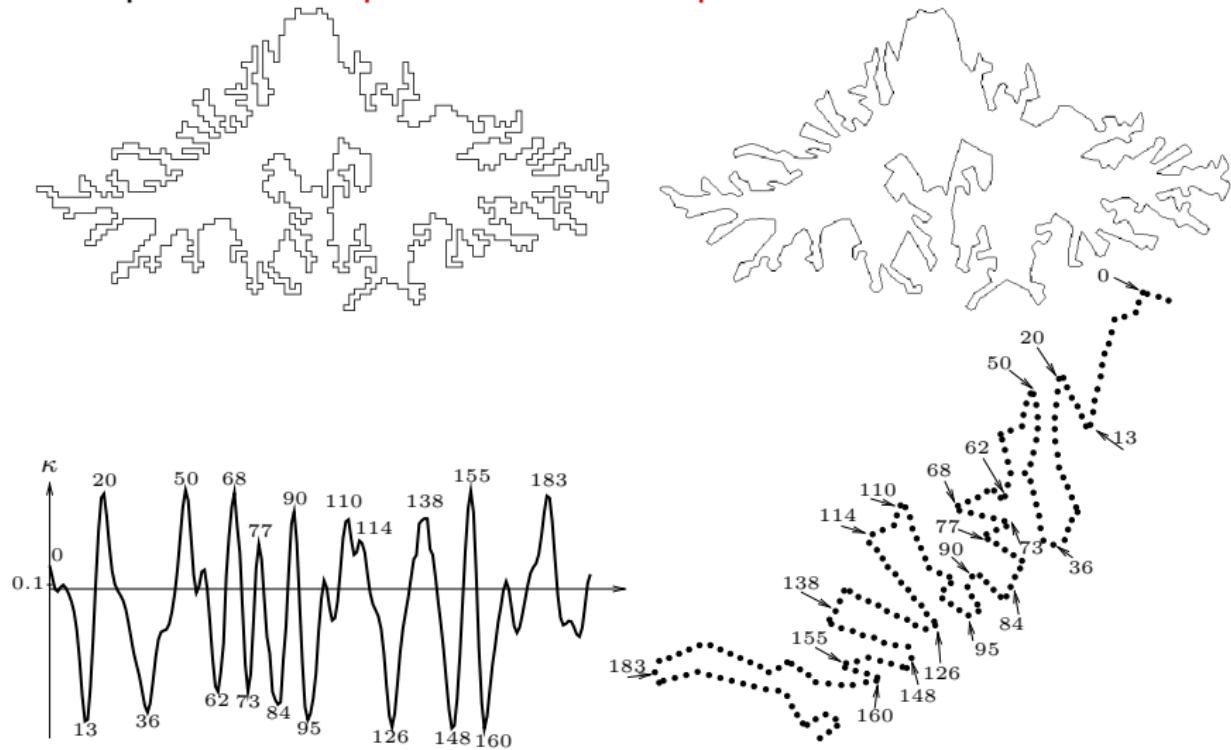
1.1107 : distance moyenne entre deux points successifs.

Taille de la fenêtre de calcul : $m = 3\sigma$, $\sigma = 3$ par exemple.

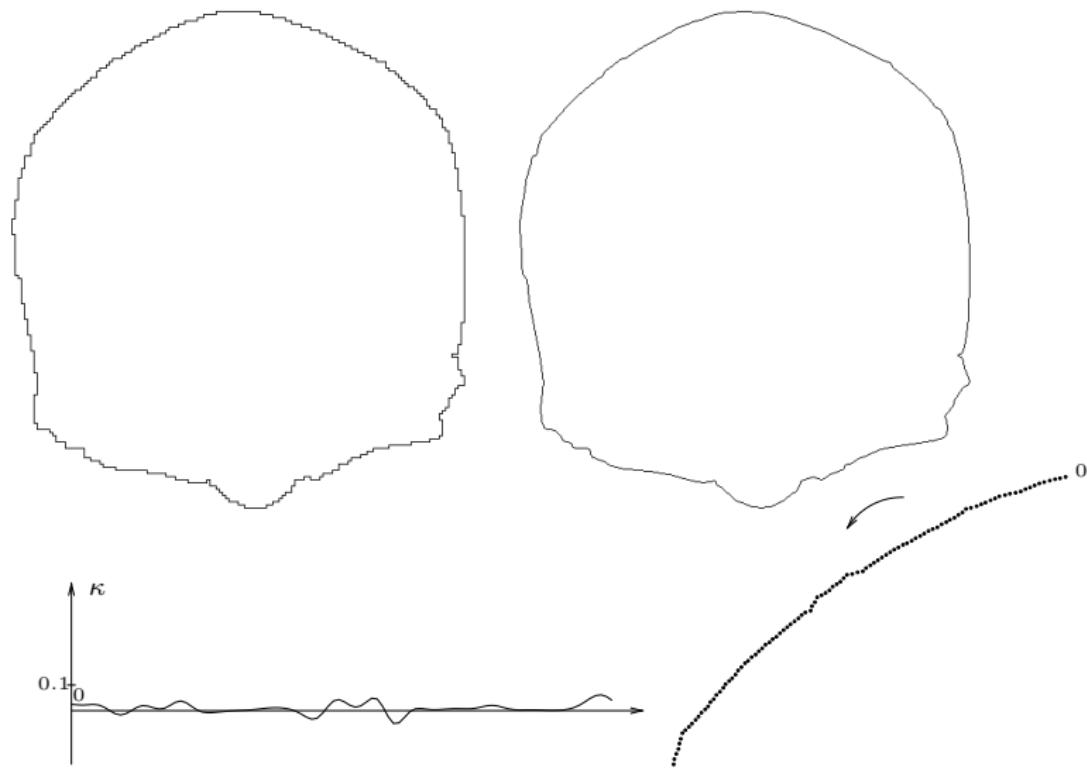
Autres méthodes de calcul de courbure : par évaluation du rayon du cercle osculateur,...

Points dominants I

Les extrema d'un profil de courbure le long d'un contour correspondent aux **points caractéristiques** du contour.



Points dominants II



3 - Caractéristiques géométriques d'une frontière discrète

Contour discret 2D : tangente

Contour discret 2D : longueur / périmètre

Contour discret 2D : courbure

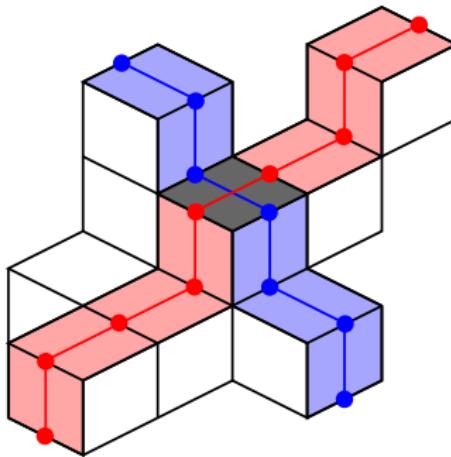
Extension en 3D

Extension 3D : vecteur normal I

Région 3D : ensemble de voxels

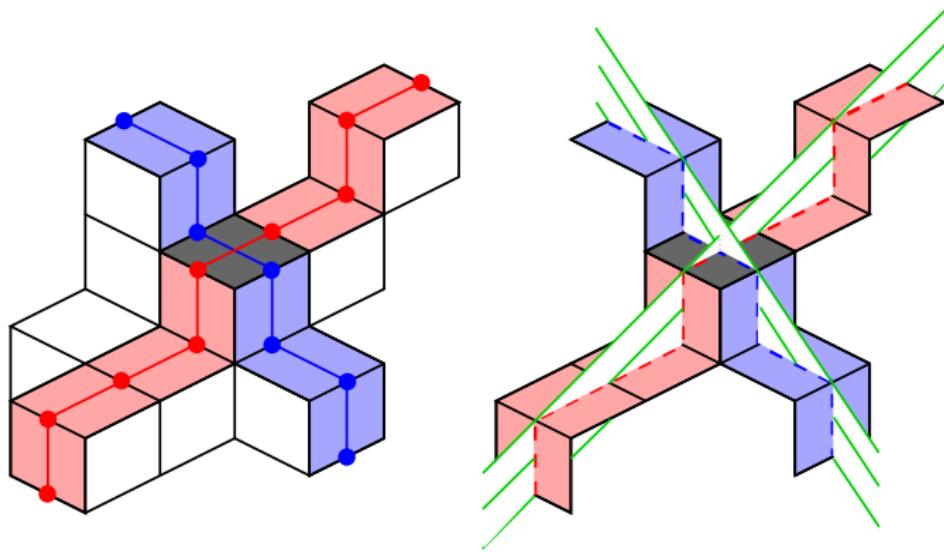
Surface d'une région 3D : ensemble de faces de voxels (surfels)

Un surfel \Rightarrow deux contours 4-connexes



Extension 3D : vecteur normal II

Calcul de la tangente sur chaque contour 2D



Directions des 2 tangentes \Rightarrow vecteur normal à la surface

Aire d'une surface I

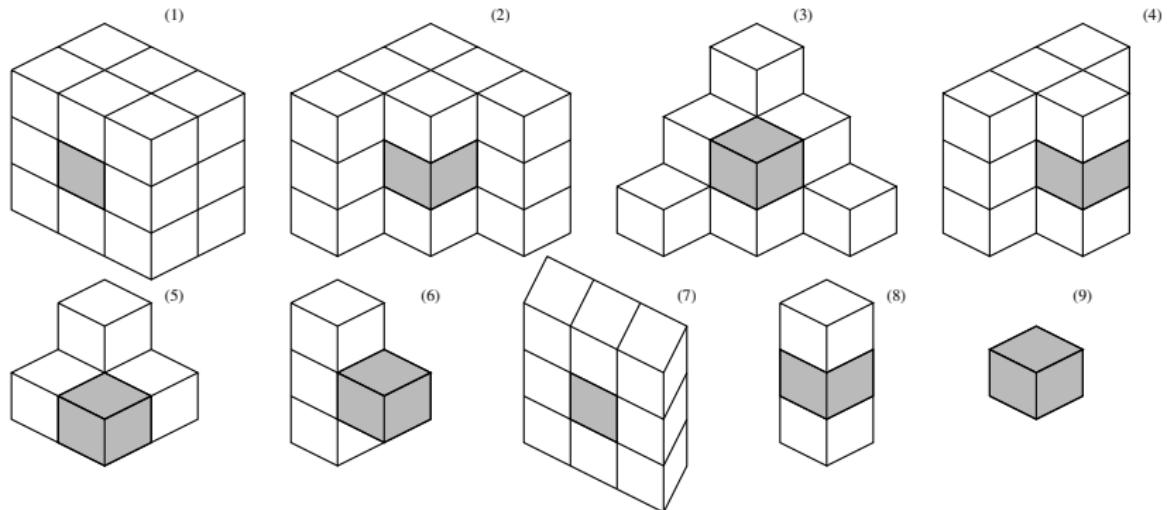
Estimateur simple [Mullikin 93]

$$\hat{S} = \sum_{i=1}^6 \psi_i N_i$$

$$\psi_1 = 0.894, \psi_2 = 1.3409, \psi_3 = 1.5879,$$

$$\psi_4 = 2, \psi_5 = \frac{8}{3}, \psi_6 = \frac{10}{3}$$

Configurations d'un voxel frontière (faces touchant le fond) :



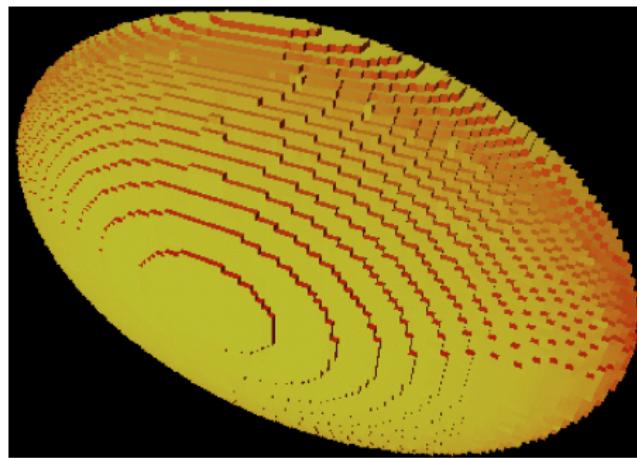
Aire d'une surface I

Intégration des normales

Contribution d'un surfel : $\vec{n} \cdot \vec{e}$

\vec{n} : vecteur normal calculé

\vec{e} : normale triviale



Sommaire

- ① Géométrie et topologie discrètes
- ② Squelettisation
- ③ Caractéristiques géométriques d'une frontière discrète
- ④ Caractéristiques géométriques d'une région

Caractériser une forme

Ici forme = région binaire 2D ou 3D

Définir des mesures :

- **robustes** au bruit
- **caractéristiques** / discriminantes
- (**invariantes** aux transformations géométriques)

→ position, dimensions, orientations, indices de forme,
mesures topologiques, signatures de contours...

Caractéristiques géométriques simples 2D

La région R est un ensemble de pixels $(x_i, y_i)_{i=1..n}$

Aire : nombre de pixels de R . Le crénelage des contours fausse peu l'approximation.

Centre de gravité : $G = (\bar{x}, \bar{y}) = (\frac{1}{n} \sum_{i=1}^n x_i, \frac{1}{n} \sum_{i=1}^n y_i)$

Diamètre : $D = \max(\text{distance}(P, Q) / P, Q \in R)$

Compacité : $\frac{4\pi \text{ Aire}(R)}{\text{Périmètre}^2(R)}$

Allongement : $\frac{\pi L_g^2(R)}{4\text{Aire}(R)}$

Degré de convexité : $\frac{\text{Aire}(R)}{\text{Aire}(\text{Enveloppe convexe}(R))}$

Caractéristiques géométriques simples 3D

La région R est un ensemble de voxels $(x_i, y_i, z_i)_{i=1..n}$

Volume : nombre de voxels de R (si l'espacement de la grille est le même dans toutes les directions)

Centre de gravité :

$$G = (\bar{x}, \bar{y}, \bar{z}) = \left(\frac{1}{n} \sum_{i=1}^n x_i, \frac{1}{n} \sum_{i=1}^n y_i, \frac{1}{n} \sum_{i=1}^n z_i \right)$$

Compacité : $\frac{36\pi \text{ Volume}^2(R)}{\text{Surface}^3(R)}$

Moments cartésiens 2D

Définition initiale : $m_{pq} = \sum_x \sum_y x^p y^q f(x, y)$

Région binaire R : $f(x, y) = 1$ si $(x, y) \in R$ et $f(x, y) = 0$ sinon

$$\Rightarrow m_{pq} = \sum_{(x,y) \in R} x^p y^q$$

Aire : m_{00}

Centre de gravité : $(\bar{x}, \bar{y}) = (\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}})$

Moments centrés : $\mu_{pq} = \sum_{(x,y) \in R} (x - \bar{x})^p (y - \bar{y})^q$

Moments centrés normalisés : $\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{\frac{p+q}{2} + 1}}$

Matrice de covariance :

$$\frac{1}{m_{00}} \begin{pmatrix} \mu_{20} & \mu_{11} \\ \mu_{11} & \mu_{02} \end{pmatrix}$$

Mesures invariantes aux rotations

$$\phi_1 = \eta_{20} + \eta_{02} \quad \phi_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2$$

$$\phi_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2$$

Axes principaux en 2D I

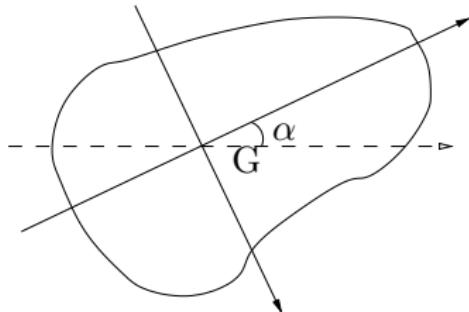
Axes principaux (directions principales) : vecteurs propres de la matrice de covariance des pixels de la région.

Matrice de covariance :

$$M_C = \begin{pmatrix} a & c \\ c & b \end{pmatrix} = \begin{pmatrix} \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2 & \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y}) \\ \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y}) & \frac{1}{n} \sum_{i=1}^n (y_i - \bar{y})^2 \end{pmatrix}$$

M_C symétrique \Rightarrow vecteurs propres orthogonaux

Axes principaux en 2D II



$$\tan(2\alpha) = \frac{2c}{a - b}$$

α angle entre l'axe des x et la 1ère direction principale

$$a = \left(\frac{1}{n} \sum_{i=1}^n x_i^2 \right) - \bar{x}^2, \quad b = \left(\frac{1}{n} \sum_{i=1}^n y_i^2 \right) - \bar{y}^2, \quad c = \left(\frac{1}{n} \sum_{i=1}^n x_i y_i \right) - \bar{x} \bar{y}$$

Remarque : on peut définir un rectangle d'encadrement de R à partir des directions principales.

Moments cartésiens 3D

Définition (région binaire) : $m_{pqr} = \sum_{(x,y,z) \in R} x^p y^q z^r$

Volume : m_{000}

Centre de gravité : $(\bar{x}, \bar{y}, \bar{z}) = \left(\frac{m_{100}}{m_{000}}, \frac{m_{010}}{m_{000}}, \frac{m_{001}}{m_{000}} \right)$

Moments centrés : $\mu_{pqr} = \sum_{(x,y,z) \in R} (x - \bar{x})^p (y - \bar{y})^q (z - \bar{z})^r$

Moments centrés normalisés : $\eta_{pqr} = \frac{\mu_{pqr}}{\mu_{000}^{\frac{p+q+r}{3} + 1}}$

Matrice de covariance : $\frac{1}{m_{000}} \begin{pmatrix} \mu_{200} & \mu_{110} & \mu_{101} \\ \mu_{110} & \mu_{020} & \mu_{011} \\ \mu_{101} & \mu_{011} & \mu_{002} \end{pmatrix}$

vecteurs propres \leftrightarrow axes principaux de la forme.

Mesures invariantes aux rotations

voir : *Geometric moment invariants, Dong Xu, Pattern Recognition 2008*

Moments de Zernike 2D

$A_{nm} = \frac{n+1}{\pi} \sum_x \sum_y f(x, y) [V_{nm}(x, y)]^*$ pour $x^2 + y^2 \leq 1$,
 n entier positif ou nul, m entier vérifiant $n - |m|$ pair et $|m| \leq n$

$V_{nm}(x, y) = R_{nm}(x, y) e^{im\arctan(\frac{y}{x})}$ base de polynômes complexes orthogonaux

$R_{nm}(x, y) = \sum_{s=0}^{\frac{n-|m|}{2}} \frac{(-1)^s (x^2 + y^2)^{\frac{n}{2}-s} (n-s)!}{s! (\frac{n+|m|}{2}-s)! (\frac{n-|m|}{2}-s)!}$ polynôme radial

$|A_{nm}|$ invariant aux rotations : soit A'_{nm} calculé après rotation d'angle θ , $A'_{nm} = A_{nm} e^{-im\theta}$

Pour obtenir l'invariance aux translations et aux changements d'échelle, il faut normaliser avec les moments cartésiens :

$$h(x, y) = f\left(\frac{x}{a} + \bar{x}, \frac{y}{a} + \bar{y}\right) \text{ avec } a = \sqrt{\frac{\beta}{m_{00}}}$$

Reconstruction : $f(x, y) = \lim_{N \rightarrow \infty} \sum_{n=0}^N \sum_m A_{nm} V_{nm}(x, y)$