

RdF – Reconnaissance des Formes

Semaine 2 : attributs de contour

Master ASE : <http://master-ase.univ-lille1.fr/>
Master Informatique : <http://www.fil.univ-lille1.fr/>
Spécialité IVI : <http://master-ivi.univ-lille1.fr/>

Plan du cours

1 – Codage de Freeman

codage de Freeman absolu

codage de Freeman relatif

variantes du codage de Freeman

2 – Codage polygonal

régression linéaire et axe principal d'inertie

approximation polygonale d'un contour

réduction d'un contour par l'algorithme de la corde

3 – Transformée de Hough

transformée de Hough: principe et méthode

recherche de droites dans une image

discrétisation de l'espace des paramètres

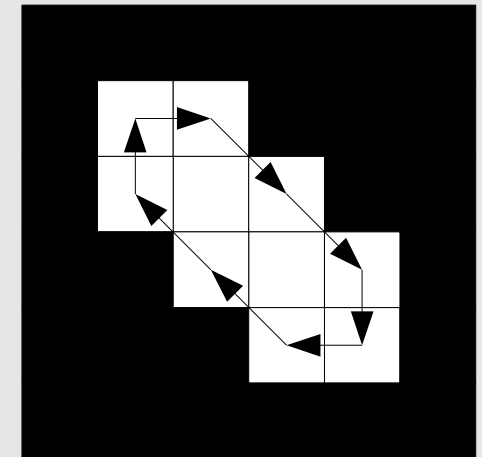
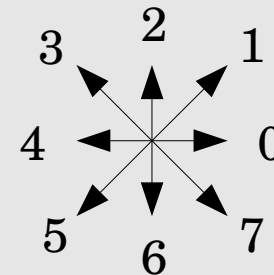
Codage de Freeman

Principe

- codage avec un nombre **limité de bits** de la direction locale d'un élément de contour défini dans une **image discrète**.
- constitution d'une **chaîne de codes** à partir d'un pixel initial, considérant qu'un élément de contour relie **2 pixels connexes**.

Méthode

- 1) on choisit un **pixel initial** du contour et un **sens de parcours**;
- 2) on code la direction qui permet de passer d'un pixel du contour à son **voisin immédiat**;
- 3) on continue jusqu'à revenir au **pixel initial**.

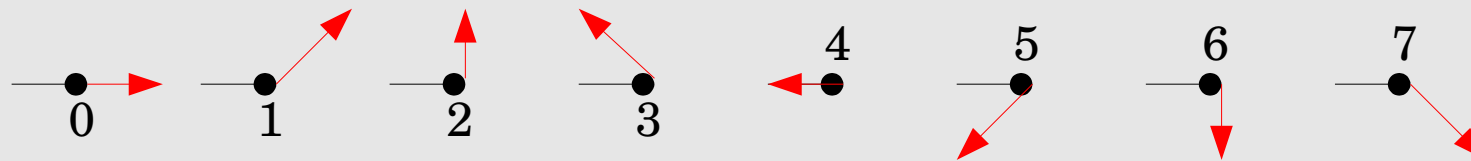


0 7 7 6 4 3 3 2

Codage de Freeman relatif

Principe

on code le **changement** de direction plutôt que la direction.



Propriétés

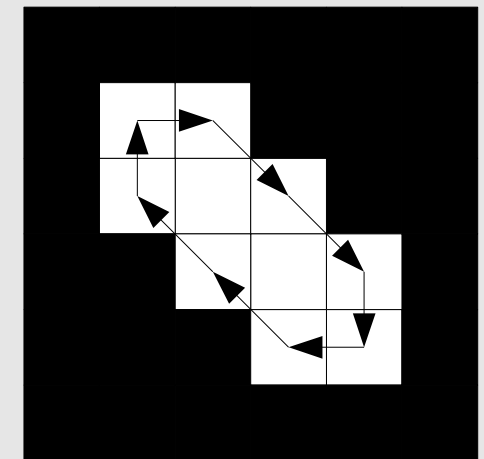
- le code de Freeman standard est invariant en **translation** uniquement.
- le code de Freeman relatif est invariant en **translation** et aux **rotations** de 45 deg.

Variantes

codage sur 2 bits pour **connexité 4**

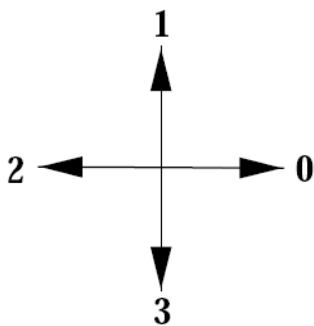
codage sur 3 bits pour **connexité 8**

codage sur 4 bits pour **connexité 8 + longueur 2**

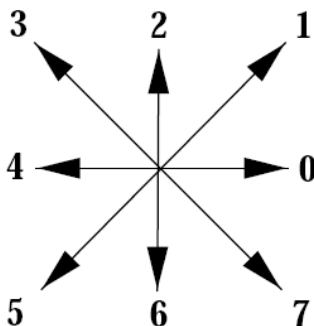


0 7 0 7 6 7 0 7

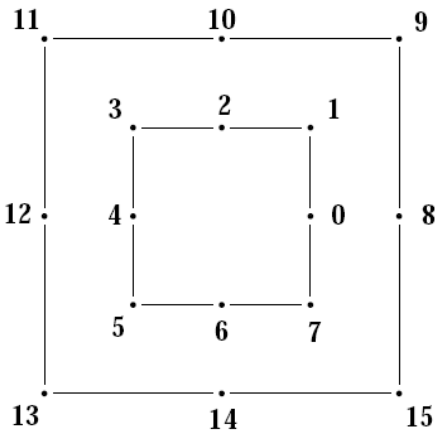
Variantes du codage de Freeman



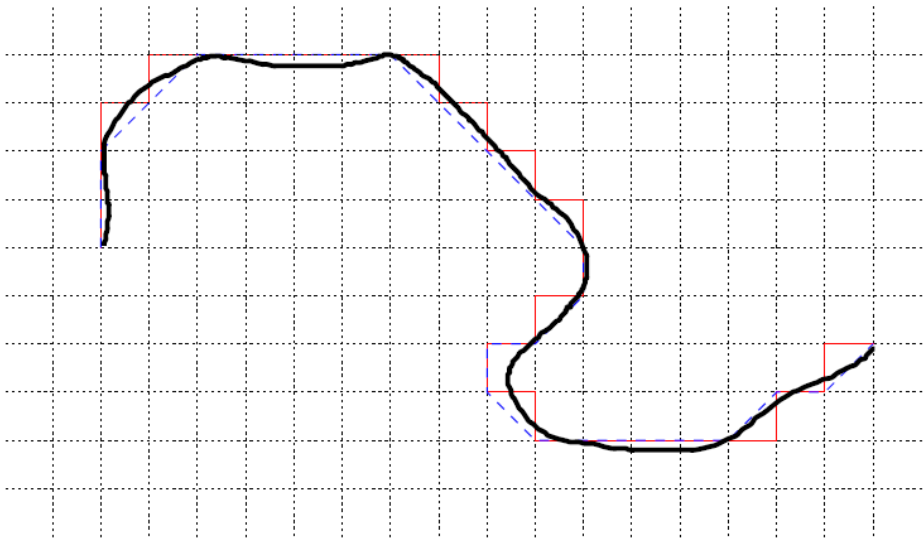
4 directions : 2 bits



8 directions : 3 bits



16 directions : 4 bits



1 1 1 0 1 0 0 0 0 0 0 3 0 3 0 3 0 3 2 3 2 3 0 3 0 0 0 0 0 1 0 1 0	34 x 2 = 68 bits
2 2 1 1 0 0 0 0 7 7 7 7 6 5 4 6 7 0 0 0 0 1 0 1	24 x 3 = 72 bits
10 9 8 8 15 15 6 5 4 6 7 8 8 1 0 1	16 x 4 = 64 bits

Régression linéaire

Principe

approcher un ensemble de points par un **segment de droite**.

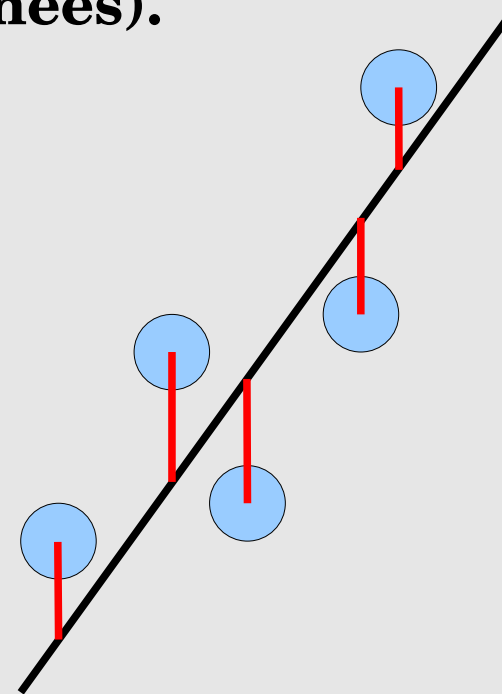
Méthode

- minimisation d'un **résidu** entre le **modèle** (la droite) et les **données** (points repérés par leurs coordonnées).
- données : n points de coordonnées (x_i, y_i)
- modèle : droite d'équation $y=a.x+b$

- **résidu** :
$$d^2(a, b) = \sum_{i=1}^n \left(y_i - (a \cdot x_i + b) \right)^2$$

Solution

$$a = \frac{\text{cov}(X, Y)}{\text{var}(X)} \quad b = E(Y) - a \cdot E(X)$$



Axe principal d'inertie

Principe

minimiser un résidu qui est la somme des **distances** au carré entre les points et la droite recherchée.

Méthode

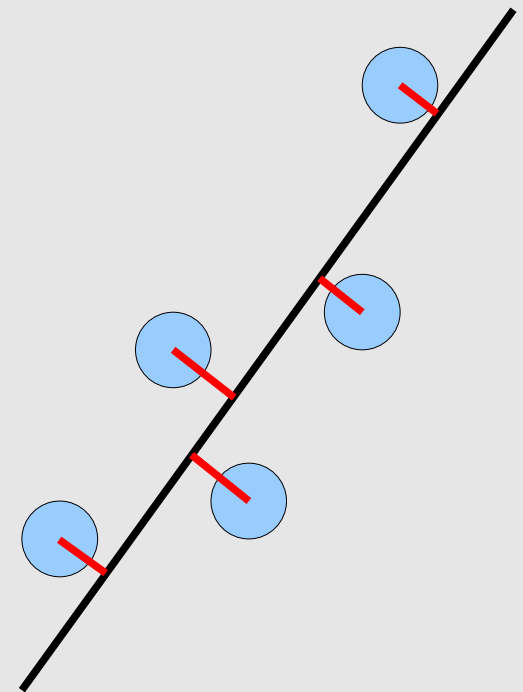
avec la contrainte $a^2+b^2=1$, le résidu prend la forme :

$$d^2(a, b) = \sum_{i=1}^n \left(a \cdot x_i + b \cdot y_i + c \right)^2 + \lambda \left(1 - (a^2 + b^2) \right)$$

Solution

l'orientation de la droite est le **vecteur propre** qui correspond à la **plus petite valeur propre** de la matrice d'inertie des points.

puis : $c = -a \cdot E(X) - b \cdot E(Y)$



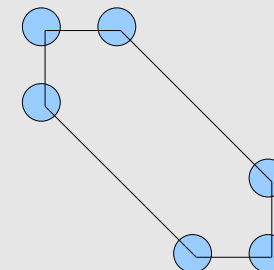
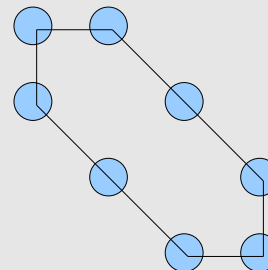
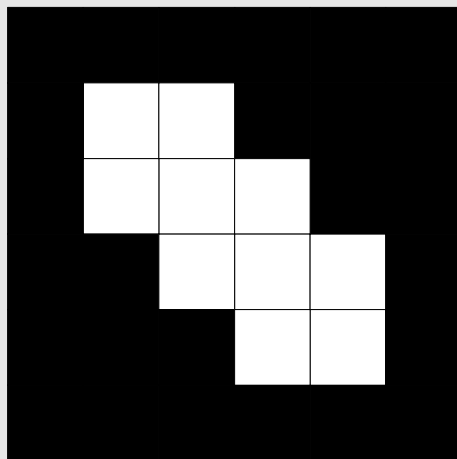
Approximation polygonale

Objectif

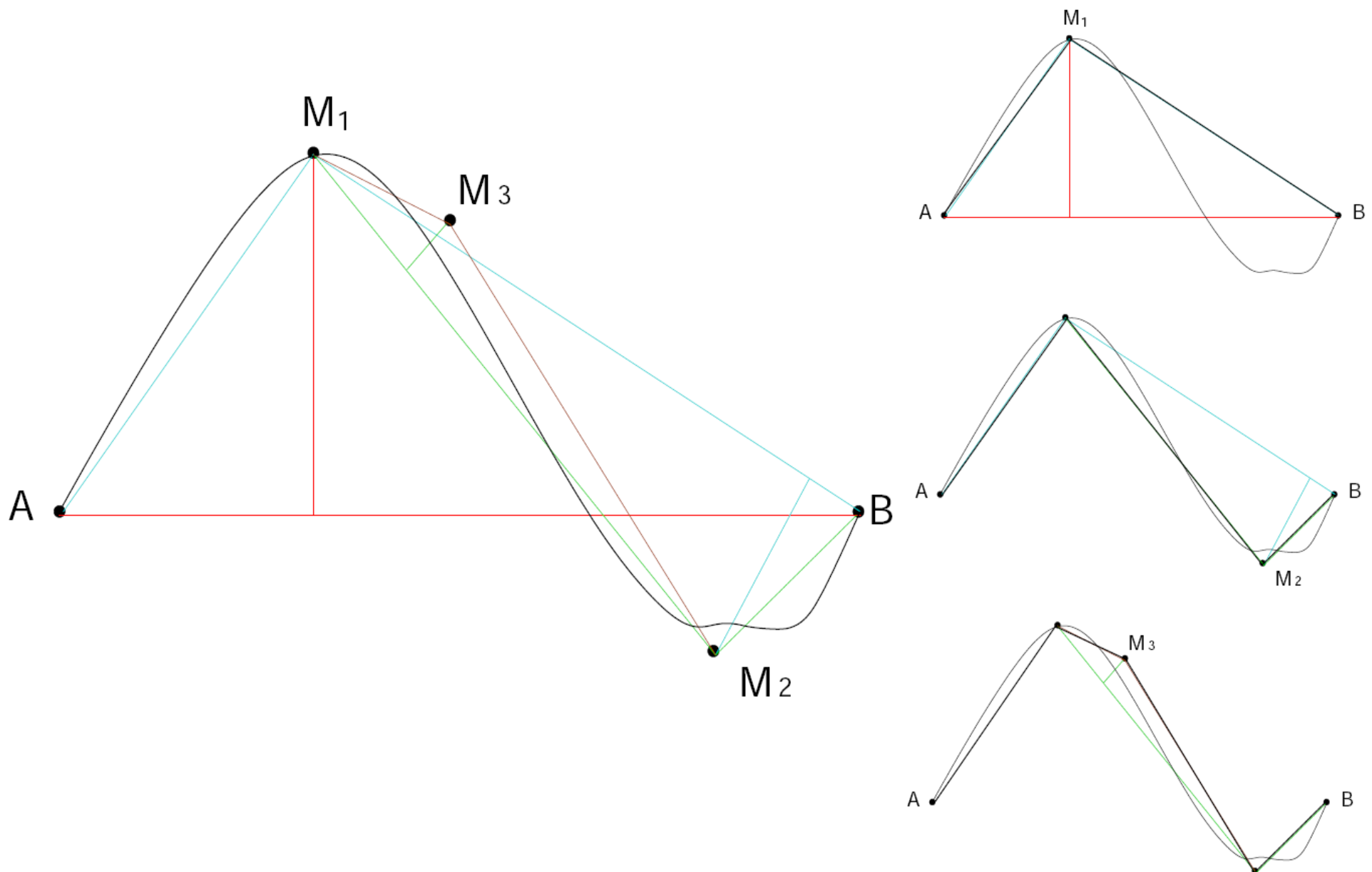
représenter le contour d'une forme par une série de **segments de droite**, eux-mêmes définis par une **série de points**.

Méthode

- détermination du contour de la forme en tenant compte de **tous** les pixels (ex: méthode similaire au codage de Freeman).
- **remplacement** des séries de points **alignés** par des segments.



Algorithme de la corde



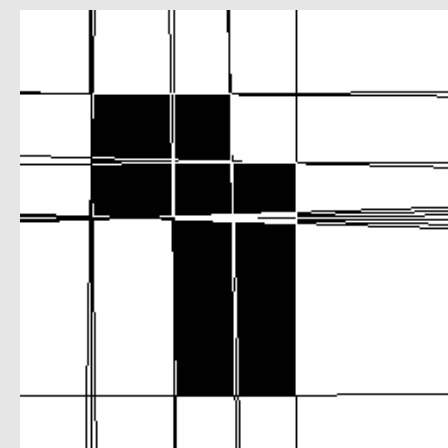
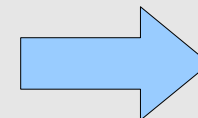
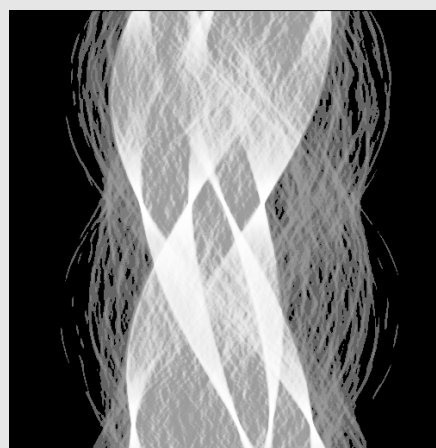
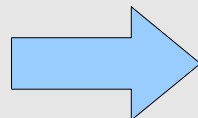
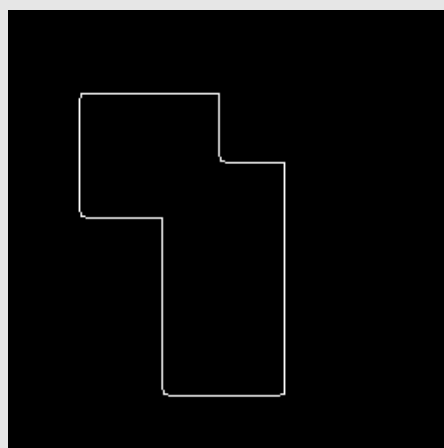
Transformée de Hough

Objectif

rechercher des courbes **paramétriques** dans une image.

Principe

- on associe à l'image (fonction binaire de deux variables réelles) un **espace transformé** dont les axes sont les **paramètres** des courbes recherchées dans l'image.
- une **courbe** dans l'image correspond à un point d'**accumulation** dans l'espace des paramètres.



Recherche de droites (1/2)

Courbe paramétrique

une droite a pour **équation paramétrique** : $y = a.x + b$

Principe de calcul de la transformée

à **chaque pixel** de l'image correspond **une droite** dans l'espace des paramètres:

soit $M_i = (x_i, y_i)$, si $y_i = a \cdot x_i + b$, alors $b = -x_i \cdot a + y_i$.

Méthode

- on balaye l'image à la recherche de **pixels non nuls**.
pour chaque pixel on **trace la droite** correspondante dans l'espace des paramètres en **accumulant** les valeurs.
- quand tous les pixels ont été balayés, on recherche les **valeurs maximales** dans l'espace des paramètres.
- à chaque maximum **correspond** une droite dans l'image.

Recherche de droites (2/2)

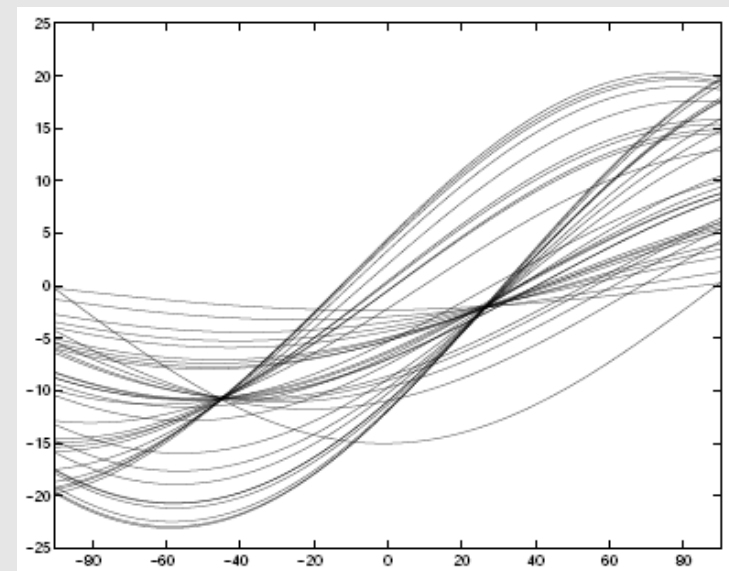
Autre paramétrisation

$$\rho = x_i \cdot \cos(\theta) + y_i \cdot \sin(\theta)$$

car l'espace des paramètres est décrit de façon **plus uniforme**.

Nouveau calcul de la transformée

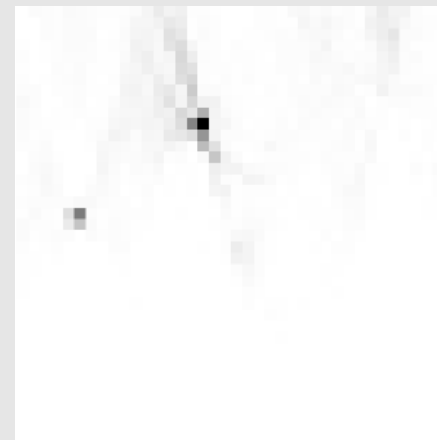
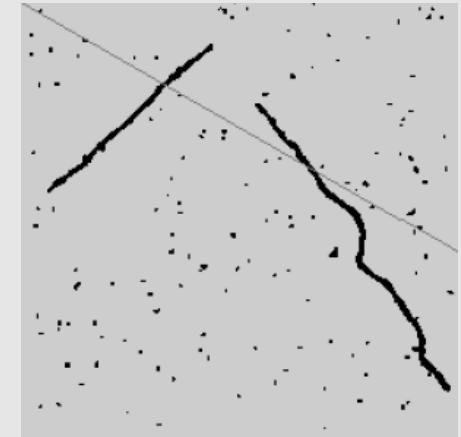
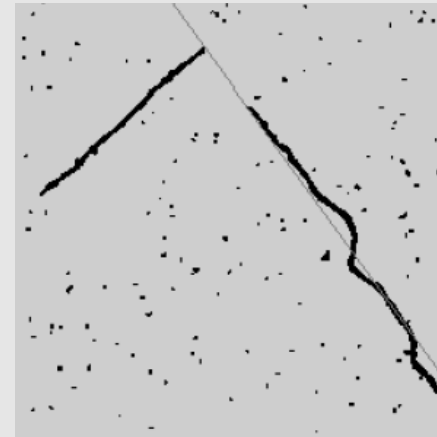
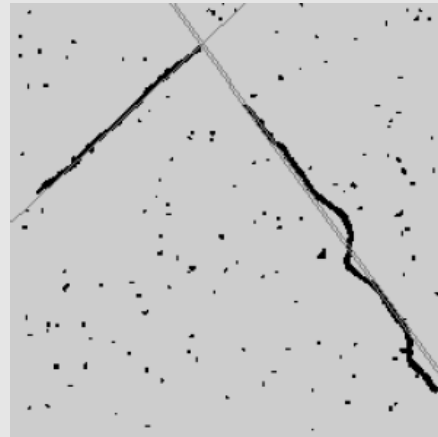
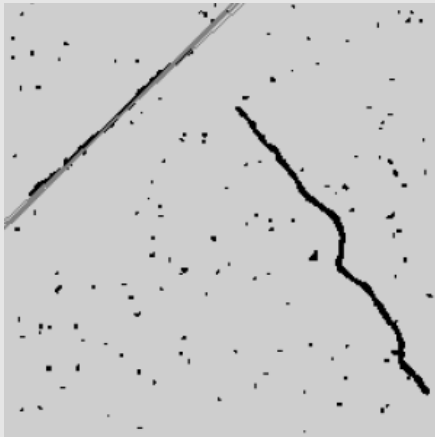
à **chaque pixel** de l'image correspond **une sinusoïde** dans le nouvel espace des paramètres.



Discrétisation de l'espace des paramètres

Intérêt

le temps de calcul est **constant** ou en tout cas **borné**.



Pour approfondir

Duda, Hart, Stork, « Pattern Classification », 2ème édition, Wiley-Interscience, 2001.

<http://rii.ricoh.com/~stork/DHS.html>

Statistical moments, Jamie Shutler (CVonline: Robert B. Fisher)

http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/SHUTLER3/CVonline_moments.html

cours de master de Florence Tupin (ENST)

http://www.tsi.enst.fr/~tupin/NEW_PAGE/cours.html