

Détection des segments dans les images digitales

Jérémie Jakubowicz

`jakubowi@cmla.ens-cachan.fr`

CMLA, ENS CACHAN

Le traitement des images digitales en quelques exemples

- Correction des défauts du système d'acquisition (aberrations, aliasing, flou)
- Compression d'images (jpeg, jpeg2000, ...)
- Débruitage (industrie photo et cinéma)
- Inpainting
- Fouille d'image
- Reconnaissance de caractère imprimés (OCR), manuscripts
- Lecture de plaques d'immatriculation (vidéo-surveillance)

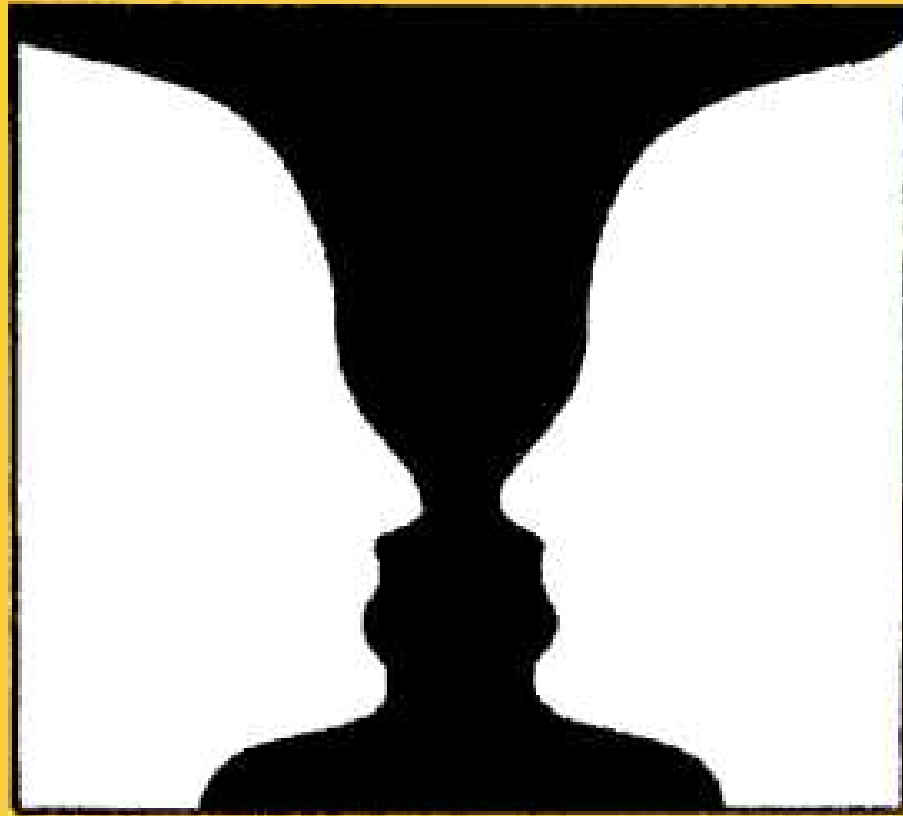
A la croisée des chemins

- Industrie du logiciel d'édition photo
- Industrie du cinéma
- Industrie de la vidéo-surveillance
- Mathématiques Appliquées
- Neuro-physiologie

Comment séparer le fond de la forme ?



Comment séparer le fond de la forme ?

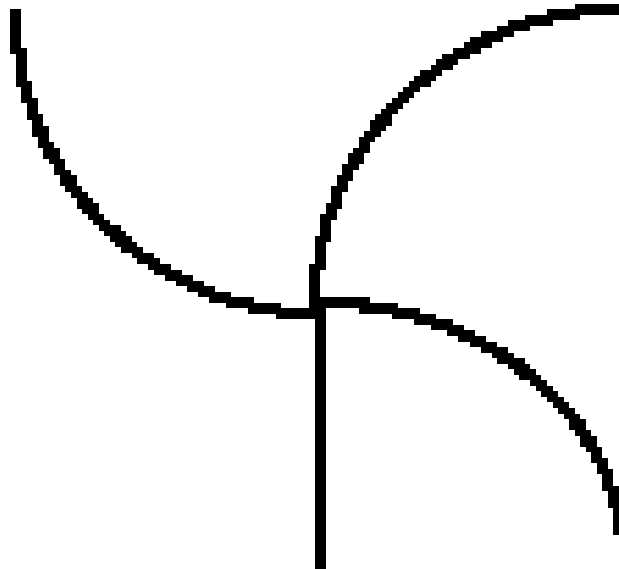


Une philosophie séduisante : l'école de la Gestalt

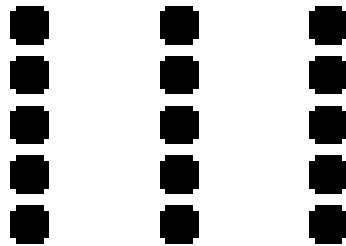
Comment faisons nous pour séparer le fond d'une image des formes qu'elle contient ?

→ On n'apprend pas tous les objets possibles mais on utilise un ensemble de règles simple de façon hiérarchique. (Gestalt)

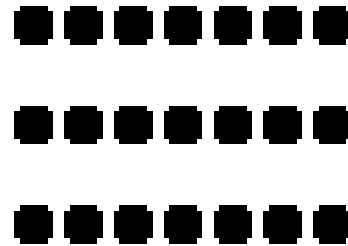
Principe de bonne continuation



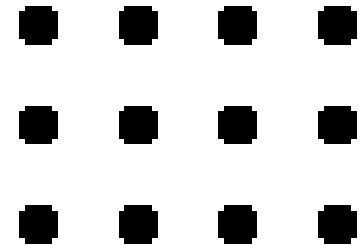
Principe de proximité



(a)

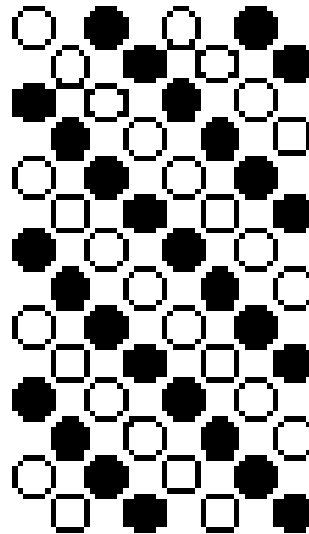


(b)

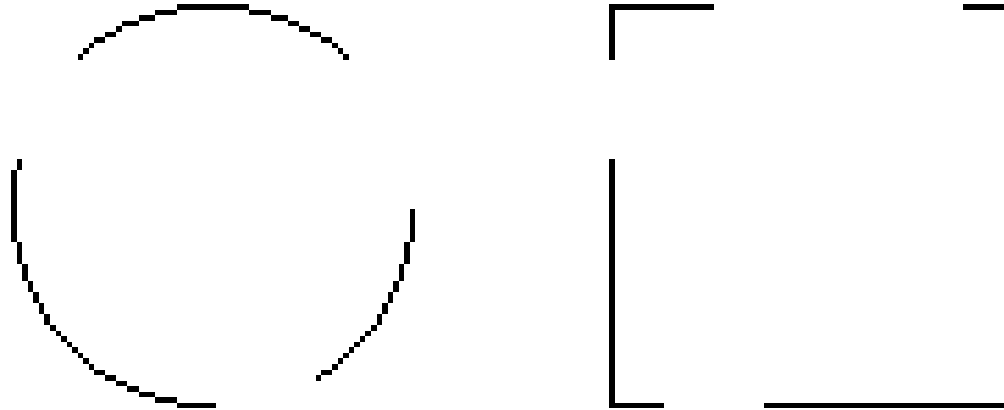


(c)

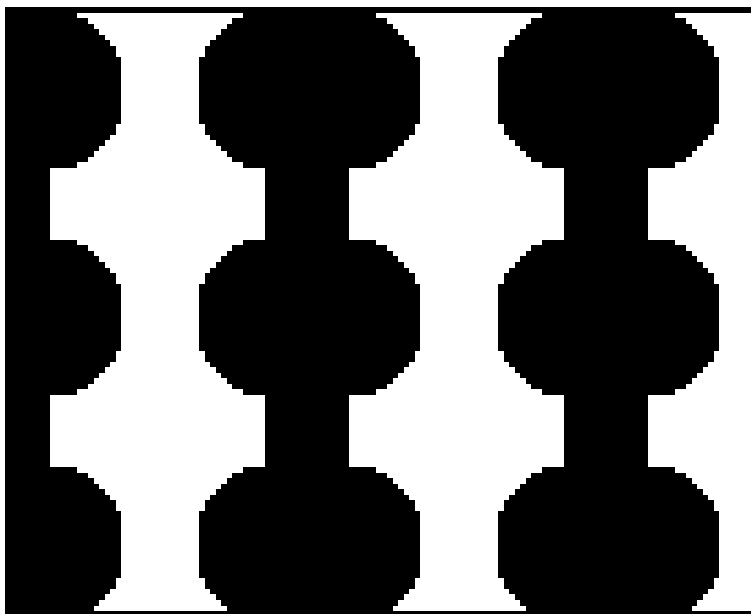
Principe de similarité (ou périodicité)



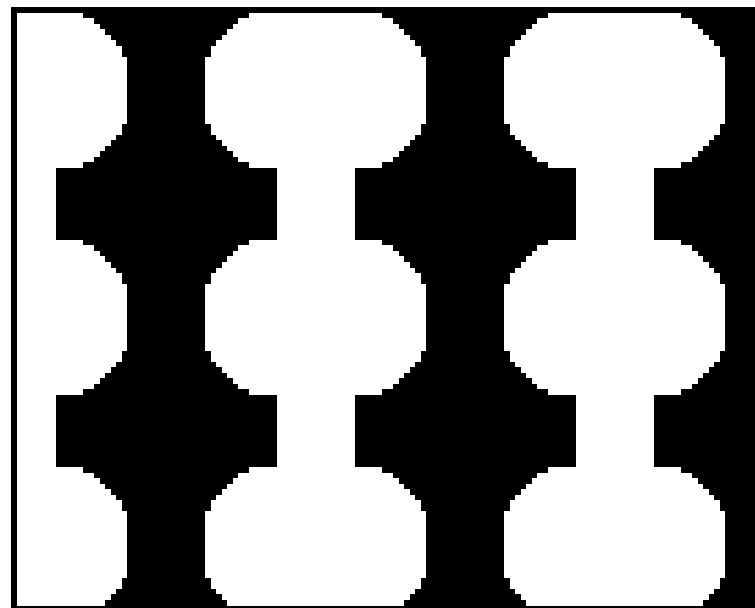
Principe de fermeture



Principe de convexité



(a)

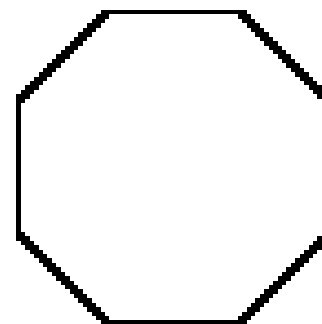
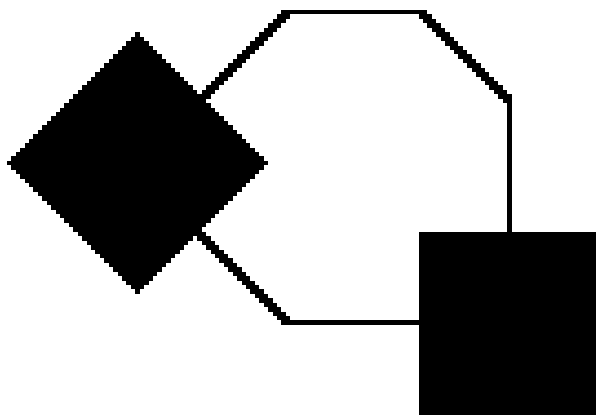
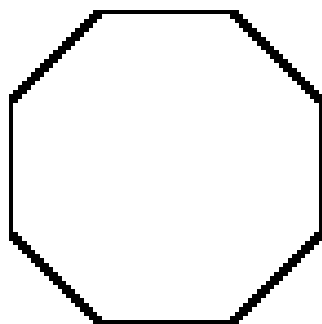


(b)

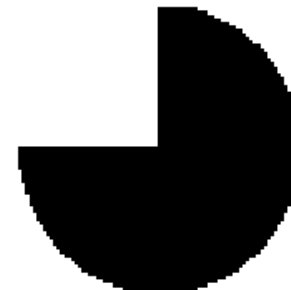
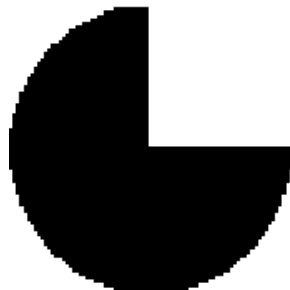
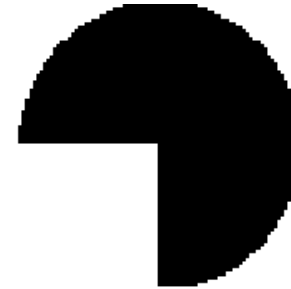
Principe de familiarité (apprentissage)

! 51C
PILASO

Principe de complétion amodale

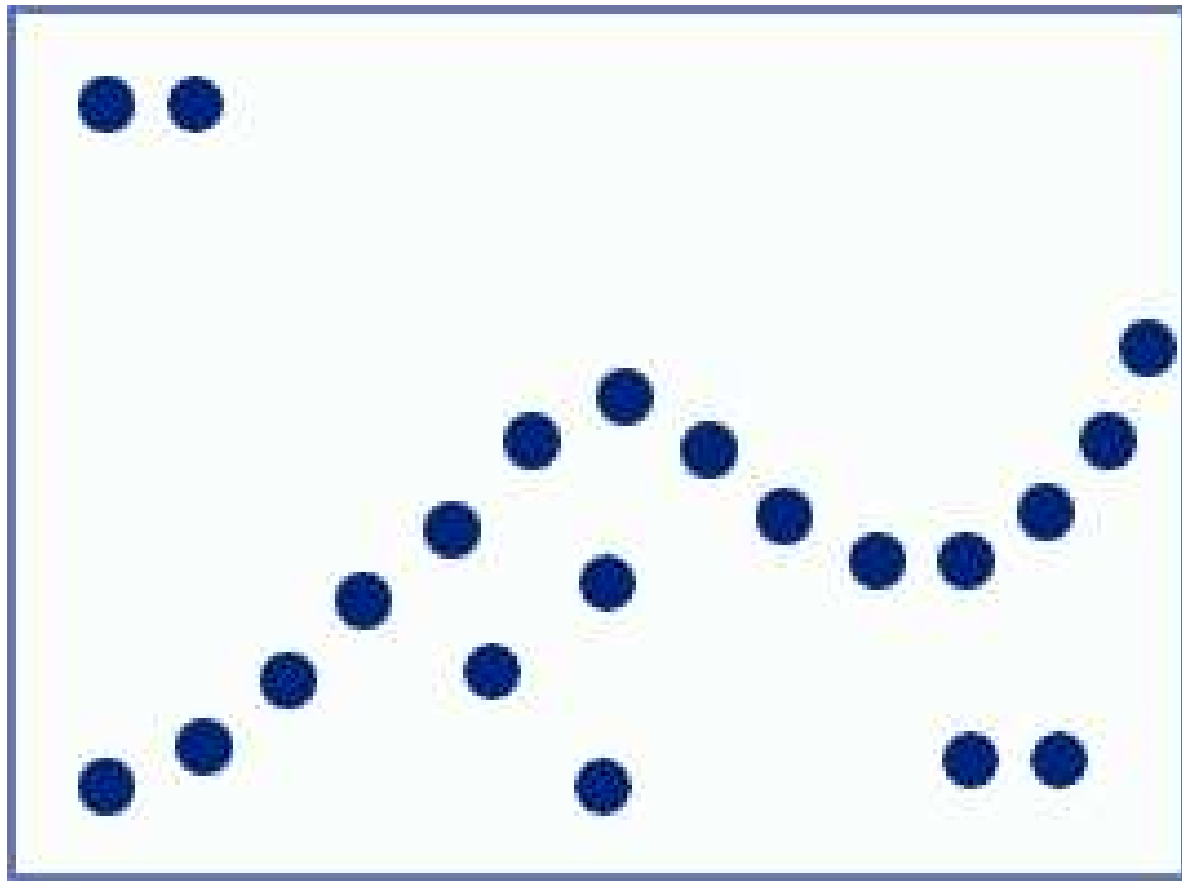


Principe de complétion modale ***(Contours illusoires)***



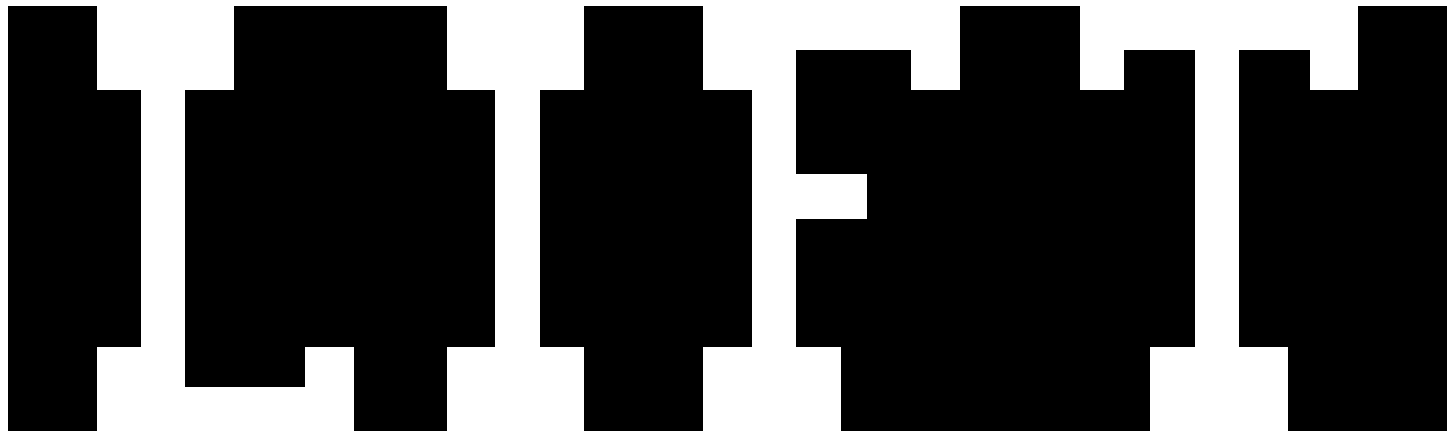
Comment combiner ces différents principes

Ces différents principes peuvent collaborer

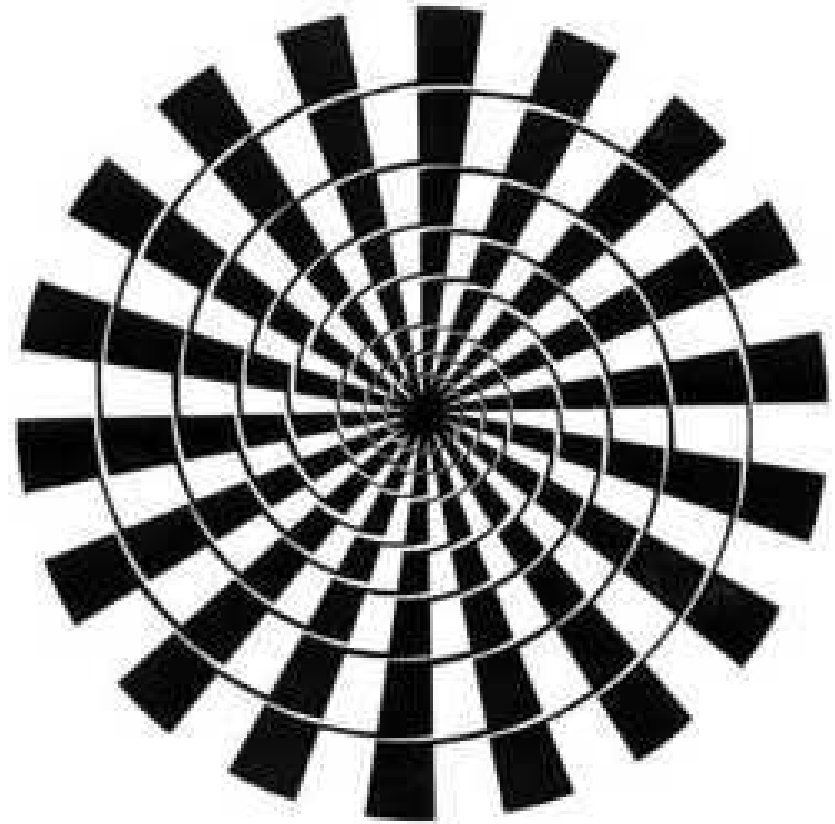


Comment combiner ces différents principes

Ou rentrer en conflit



Comment combiner ces différents principes



Ou rentrer en conflit

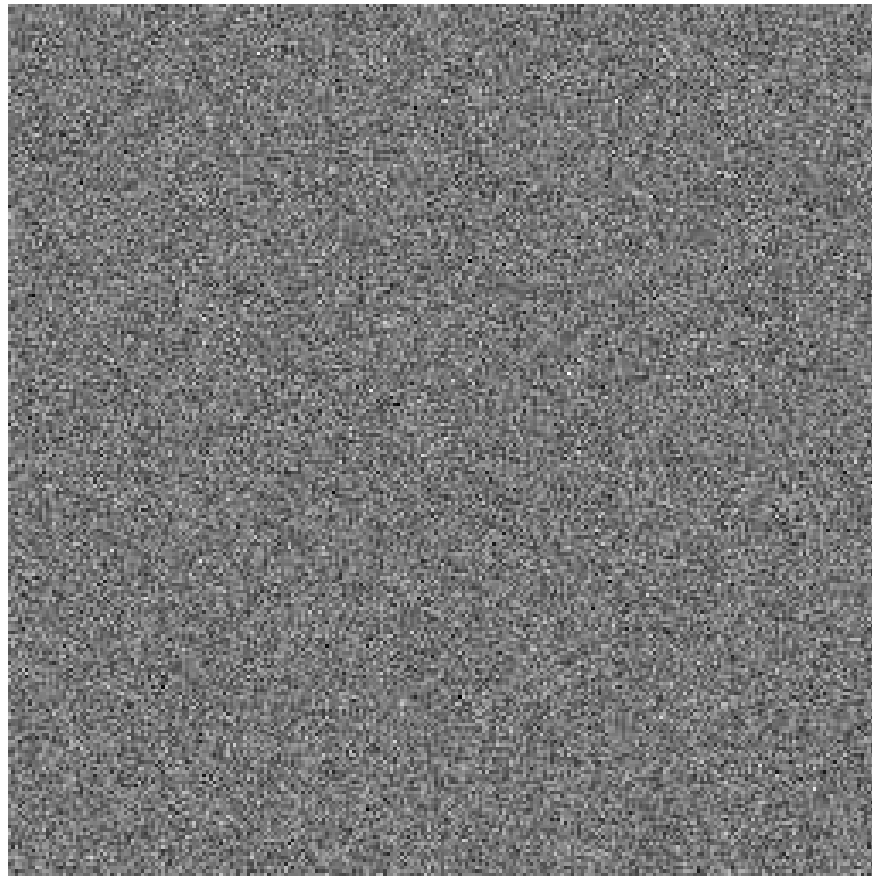
Et là, vous voyez quoi ?



Et là, vous voyez quoi ?

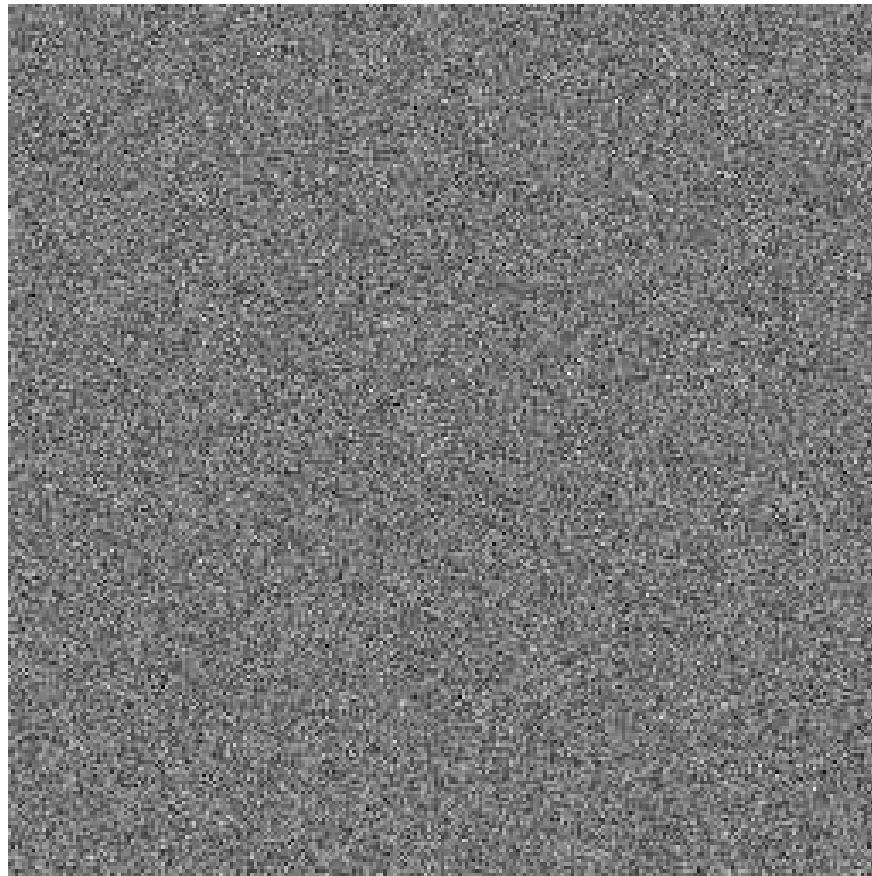


Un principe général



Un principe général

On ne perçoit pas de structure dans le bruit.



Utilité de ce principe ?

Peut-on utiliser ce principe pour construire des détecteurs de Gestalt électroniques ?

Exemple : détection de carrés

On veut détecter des carrés noirs dans un monde en noir et blanc.

Exemple : détection de carrés

On veut détecter des carrés noirs dans un monde en noir et blanc.

- Premier “ingrédient” : la description des détecteurs.

Exemple : détection de carrés

On veut détecter des carrés noirs dans un monde en noir et blanc.

- Premier “ingrédient” : la description des détecteurs.
 - On détecte un carré noir si tous les pixels qui le constituent sont noirs.

Exemple : détection de carrés

On veut détecter des carrés noirs dans un monde en noir et blanc.

- Premier “ingrédient” : la description des détecteurs.
 - On détecte un carré noir si tous les pixels qui le constituent sont noirs.
- Deuxième “ingrédient” : un modèle de bruit.

Exemple : détection de carrés

On veut détecter des carrés noirs dans un monde en noir et blanc.

- Premier “ingrédient” : la description des détecteurs.
 - On détecte un carré noir si tous les pixels qui le constituent sont noirs.
- Deuxième “ingrédient” : un modèle de bruit.
 - Tous les pixels sont noirs ou blancs indépendamment les uns des autres avec une probabilité $1/2$.

Utilisation du principe de Helmholtz

On utilise le principe précédent “on ne détecte pas de structures dans le bruit” pour calculer les paramètres de détection des carrés précédents.

Utilisation du principe de Helmholtz

On utilise le principe précédent “on ne détecte pas de structures dans le bruit” pour calculer les paramètres de détection des carrés précédents.

Combien de détection sont susceptibles d’avoir lieu en moyenne dans le modèle de bruit ?

Détection de carrés (suite)

- Suivant le modèle de bruit utilisé, la probabilité qu'un carré soit tout noir dépend uniquement de la longueur de son côté n .

$$\left(\frac{1}{2}\right)^{n \times n}$$

- Le nombre de carrés détectés (toutes tailles confondues) vaudra donc, pour une image de taille $N \times N$

$$\sum_{n=1}^N (N - n + 1)^2 \left(\frac{1}{2}\right)^{n \times n}$$

Détection de carrés (suite)

On obtient les résultats suivants, pour $N = 128$.

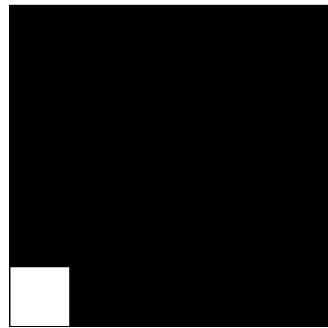
seuil	1	2	3	4	5	6
NFA	9231.31	1039.31	31.25	0.24	0.0	0.0

Ainsi il apparait que le nombre de carrés de longueur de côté plus grande que 5 est inférieure à 10^{-3} .

On peut donc fixer ce seuil pour dire observer un carré dans une image digitale de taille 128×128 .

Détection de carrés (suite)

L'algorithme précédent présente un inconvénient.



Lequel ?

Un autre détecteur de carrés

On va utiliser le même modèle de bruit. Mais d'autres détecteurs de carrés.

Le principe de Helmholtz ne dit pas tout sur la façon de construire les détecteurs (il permet d'ajuster leurs paramètres)!

Nos détecteurs choisis seront de la forme : le nombre de pixels noirs dans un carré C donné est supérieur à un seuil k_C .

Un autre détecteur de carrés (suite)

Le principe de Helmholtz nous dit que les seuils k_C doivent être choisis de façon que le nombre de carrés détectés dans le modèle de bruit soit faible.

Un autre détecteur de carrés (suite)

Le principe de Helmholtz nous dit que les seuils k_C doivent être choisis de façon que le nombre de carrés détectés dans le modèle de bruit soit faible.

- Pour un carré donné la probabilité d'une détection dans le modèle de bruit vaut

$$b(|C|, k_C) = \sum_{i=k_C}^{|C|} \binom{|C|}{i} \left(\frac{1}{2}\right)^{|C|}$$

Un autre détecteur de carrés (suite)

Le principe de Helmholtz nous dit que les seuils k_C doivent être choisis de façon que le nombre de carrés détectés dans le modèle de bruit soit faible.

- Pour un carré donné la probabilité d'une détection dans le modèle de bruit vaut

$$b(|C|, k_C) = \sum_{i=k_C}^{|C|} \binom{|C|}{i} \left(\frac{1}{2}\right)^{|C|}$$

- Au total ce nombre vaut donc

$$\sum_{n=1}^N (N - n + 1)^2 b(|C|, k_C)$$

Un autre détecteur de carrés (fin)

On ajuste les seuils de façon que le nombre de carrés détectés en moyenne dans une image de bruit (Nombre de Fausses Alarmes) soit plus petit que 10^{-3} pour une image de taille 128×128 .

Si pour tous les carrés C on a

$$b(|C|, k_C) \leq \frac{10^{-3}}{\sum_{n=1}^N (N - n + 1)^2}$$

Ce qui nous permet d'en déduire les seuils k_C .

côté	1	2	3	4	5	6	7	8	9	10	11
k_C	2	5	10	17	26	35	45	56	67	80	93

Détecter les segments

- Trouver les détecteurs adéquats

Détecter les segments

- Trouver les détecteurs adéquats
- Donner un modèle de bruit dans lequel les calculs sont possibles.

Détecter les segments

- Trouver les détecteurs adéquats
- Donner un modèle de bruit dans lequel les calculs sont possibles.
- Ajuster les seuils par le principe de Helmholtz.

Détecter les segments

- Trouver les détecteurs adéquats
- Donner un modèle de bruit dans lequel les calculs sont possibles.
- Ajuster les seuils par le principe de Helmholtz.

Cliquez ici pour commencer

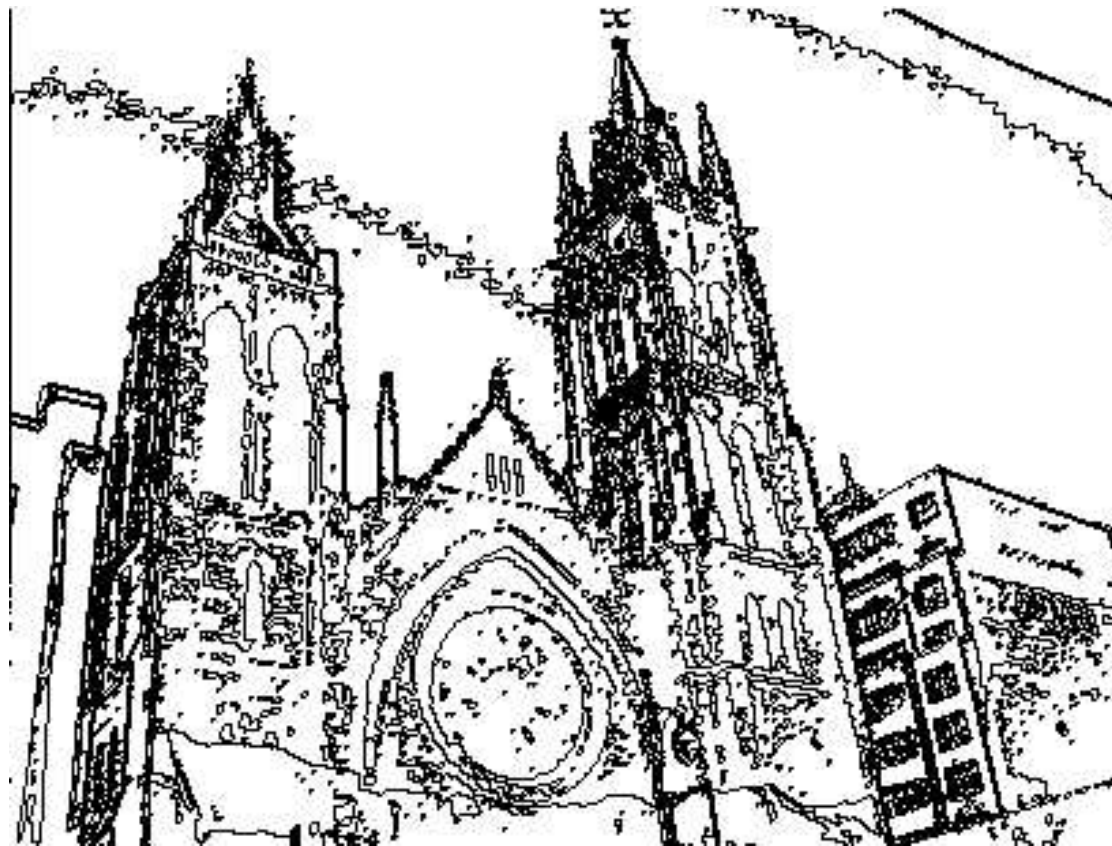
Transformation de l'image de base

Calcul du gradient de l'image.



Transformation de l'image de base

Calcul du gradient de l'image.

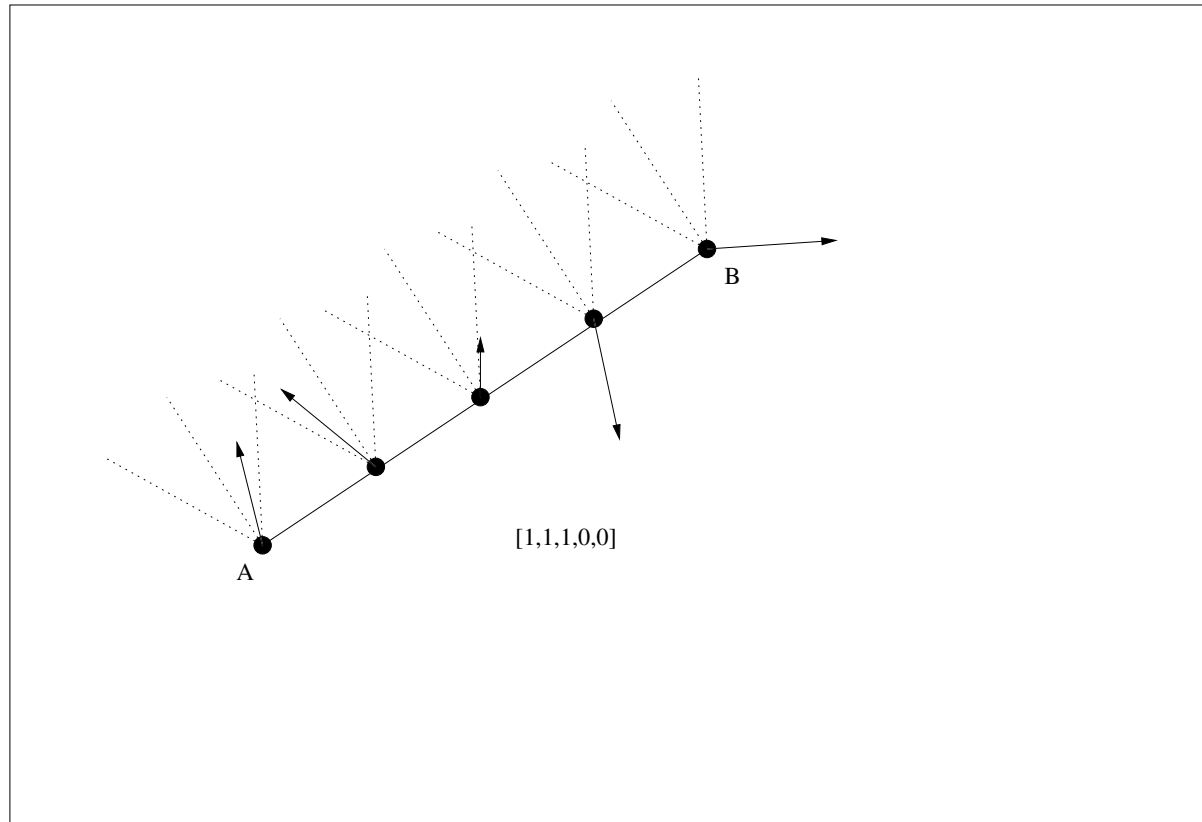


Transformation de l'image de base

Calcul du gradient de l'image.



Qu'est-ce qu'un alignement ?



La forme d'un détecteur de segment

Etant donné deux points sur l'image A et B on dira détecter un segment entre A et B si il y a plus que $k_{A,B}$ points alignés.

On choisit comme modèle de bruit, le modèle où les angles de tous les gradients de l'image sont indépendants et identiquement distribués de façon uniforme entre $]-\pi, \pi[$. On peut montrer que c'est le cas si l'image est la réalisation d'un bruit blanc gaussien.

Utilisation du principe de Helmholtz

La probabilité d'avoir au moins $k_{A,B}$ points bien alignés vaut

$$b(|AB|, k_{A,B}, p) = \sum_{k \geq k_{A,B}} \binom{|AB|}{k} p^k (1-p)^{|AB|-k}$$

Le nombre de segments détectés en moyenne “par hasard” (c'est-à-dire dans une image de bruit) vaut alors

$$\sum_{A,B} b(|AB|, k_{A,B}, p)$$

On choisit donc les seuils $k_{A,B}$ suivant :

$$b(|AB|, k_{A,B}, p) \leq \frac{\varepsilon}{N^4}$$

Objets géométriques ...

- point ou vecteur : *class Point*
 - deux coordonnées x et y
 - méthodes pour ajouter, multiplier des vecteurs, calculer une norme, un produit scalaire, un angle (ou orientation)
- segment : *class Segment*
 - deux extrémités a et b de type *Point*
 - méthodes pour calculer un vecteur directeur et un vecteur normal

... et statistiques

calcul des probabilités de lois binomiales $B(n, p)$
(fonction `tabule_queue_binom(n, p)`)

- pour n aussi grand que la plus grande dimension de l'image
- pour p : probabilité que l'angle entre deux vecteurs $< \frac{2\pi}{32} = \frac{1}{16}$
- calcul par récurrence : $a_m^k = pa_{m-1}^{k-1} + (1-p)a_{m-1}^k$
où $a_m^k = P(X = k)$ et X suit une loi binomiale $B(m, p)$

Fonctionnement du programme informatique : trois étapes

1. Préliminaires

- chargement de l'image
- affichage de l'image
- calcul du gradient : matrice de vecteurs,
 $grad[(x, y)] = \text{gradient au pixel } (x, y)$, le gradient est la seule information manipulée par l'algorithme

2. Algorithme de détection de segments

3. Affichage du résultat : une image sur laquelle on superpose les segments détectés

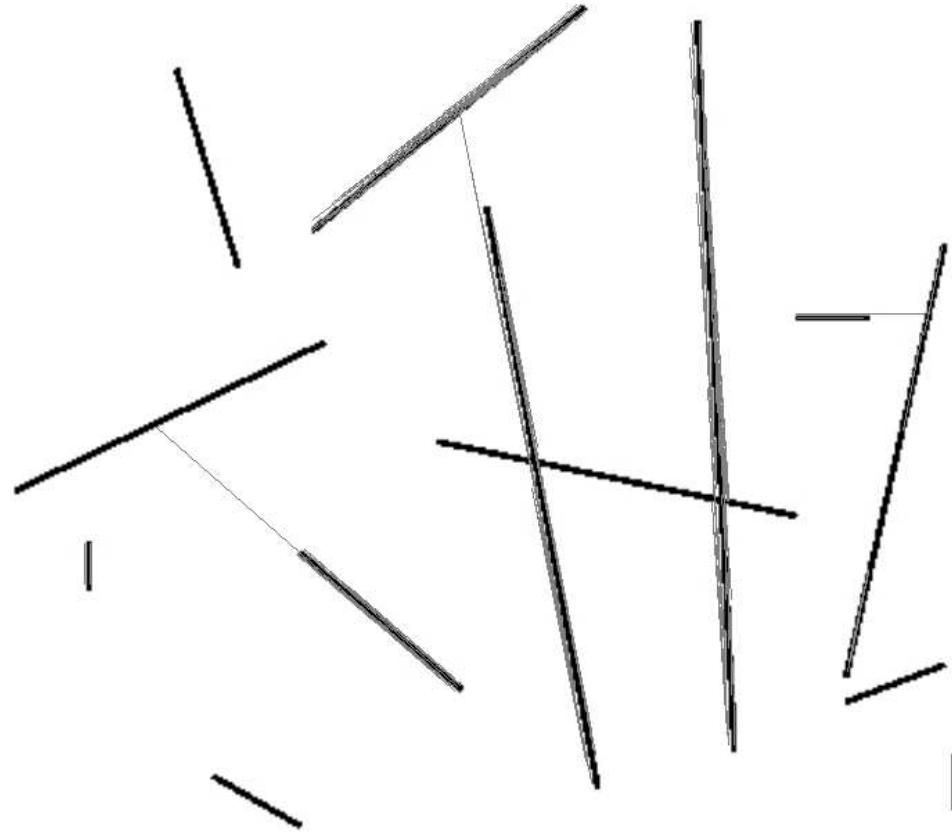
Algorithme de détection de segments

1. choix d'un segment : *class SegmentBord*
2. est-ce un segment significatif ?
 - (a) pour chaque pixel : le gradient est-il presque perpendiculaire au segment → 0 ou 1
segment → liste de 0 ou 1 (exemple : [0, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 0], stockés dans une instance de la classe *LigneGradient*)
class SegmentBord_Commune, méthode *decoupe_gradient*
 - (b) y a -t-il suffisamment de 1 pour en faire un segment ou est-ce le hasard ?
→ fait intervenir les probabilités
class LigneGradient, méthode *segments_significatifs*
 - (c) on réitère ce procédé pour chaque sous-liste
3. on recommence les étapes précédentes pour chaque segment reliant deux bords différents de l'image

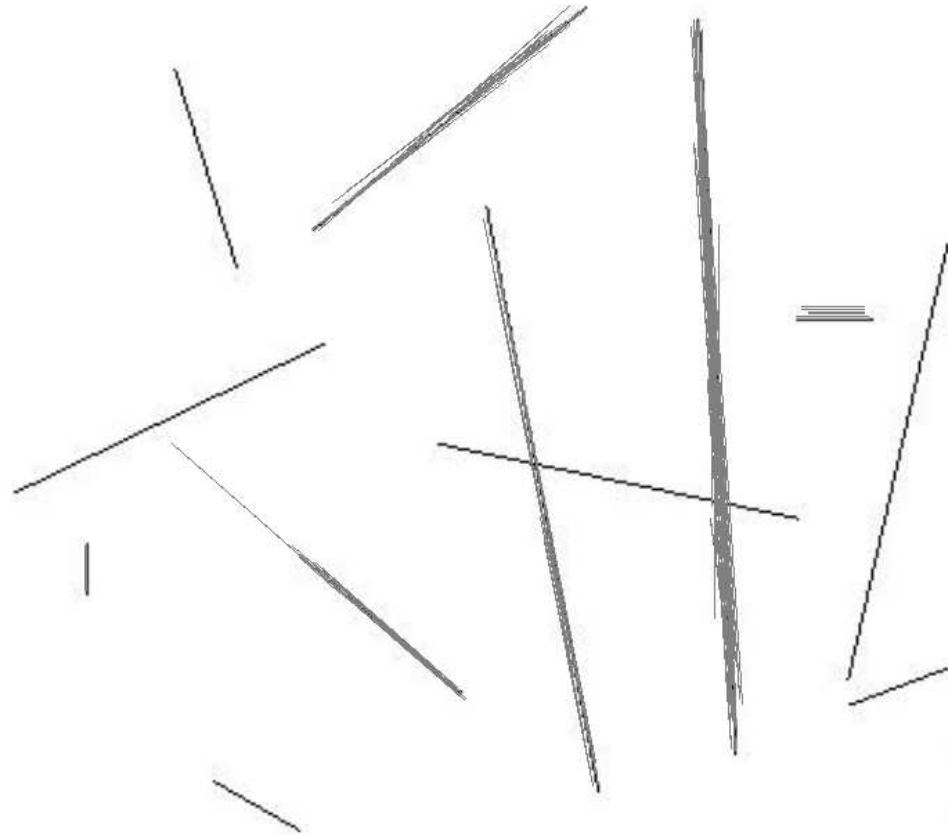
Les segments reliant deux bords différents de l'image

- 2 paramètres
 1. une orientation : un angle choisi dans l'ensemble $\{0, \delta, 2\delta, 3\delta, \dots, 2\pi - \delta\}$
 2. 1 extrémité : tout pixel du contour
- Comment les étudier tous ?
 1. choisir un premier point du contour a et un angle α nul
 2. détecter si ce segment ou une sous-partie est significatif dans l'image
 3. on passe au segment suivant : (next)
 - on change la première extrémité : on passe au pixel suivant, on déduit la seconde avec l'orientation
 - si on a fait le tour, on change l'orientation
 - si on a aussi fait le tour, on a fini

Analyse de l'algorithme sur deux exemples



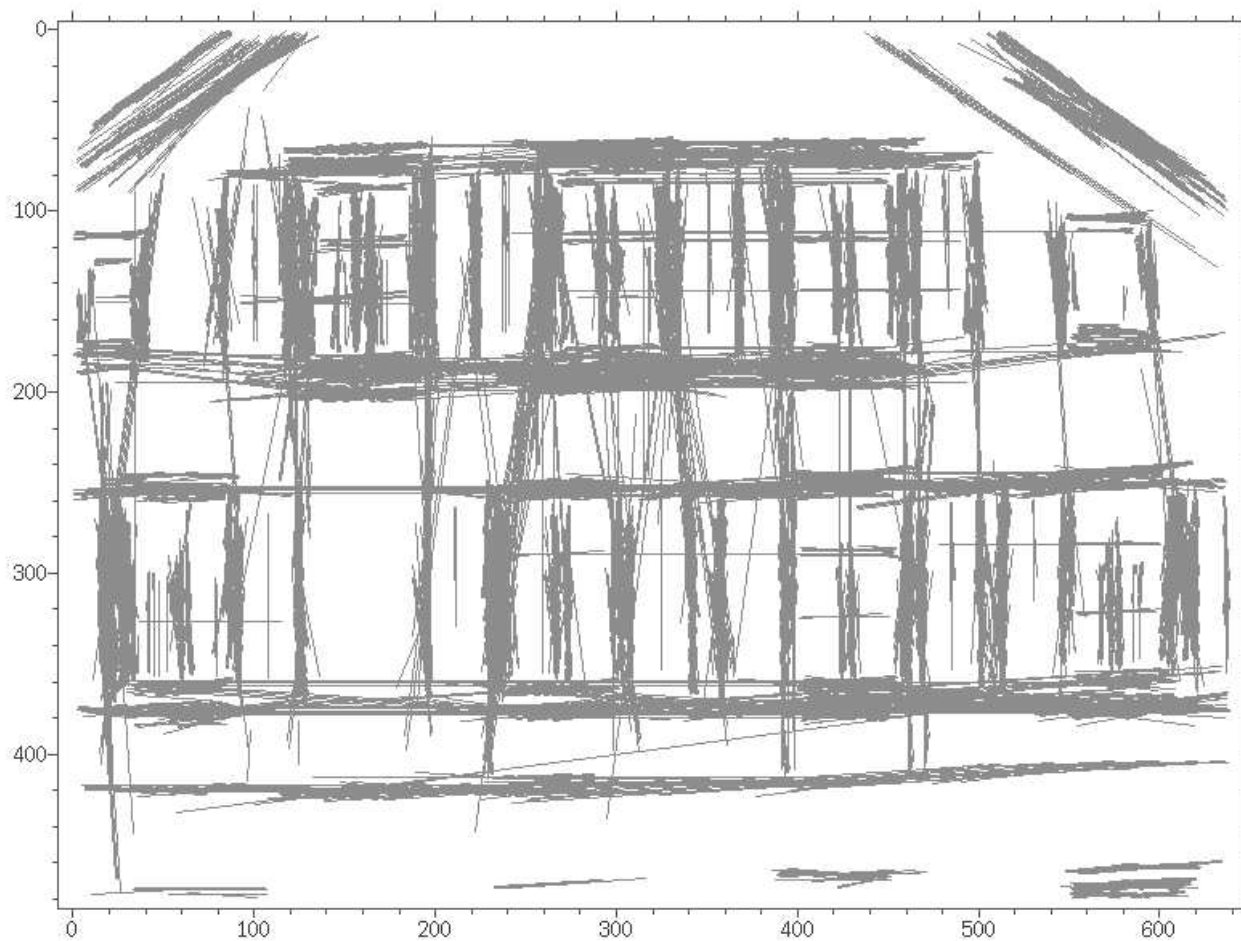
Analyse de l'algorithme sur deux exemples



Faire un croquis d'une image



Faire un croquis d'une image



Conclusion

- Les idées simples peuvent donner de bons résultats.
- Python nous a permis de faire des allers–retours entre le monde des idées et l'expérimentation numérique avec un coût raisonnable (quelques centaines de lignes de code)

Conclusion

- Les idées simples peuvent donner de bons résultats.
- Python nous a permis de faire des allers–retours entre le monde des idées et l'expérimentation numérique avec un coût raisonnable (quelques centaines de lignes de code)

Pour aller plus loin

Lire [1].

References

- [1] A. Desolneux, L. Moisan, and J.-M. Morel.
Computational gestalt theory, september 2003.

<http://www.cmla.ens-cachan.fr/Utilisateurs/morel/lecturenote.pdf>