

Traitement d'images

Image Processing

DANIEL RACOCEANU
PROFESSEUR, SORBONNE UNIVERSITÉ



1

Traitement d'images : construction de l'UE (1/3)

Niveau du cours : M1 - Semestre 2

Nombre de crédits : 3 ECTS

Code : MU4RBI07 / MU4RBI08

1. Objectifs pédagogiques :

Techniques essentielles de traitement d'images.

2. Compétences acquises :

Comprendre et maîtriser les algorithmes et méthodes les plus importantes en traitement d'image

Capacité de trouver des solutions / instantiations opérationnelles pour les applications / projets abordés.

2



Traitement d'images : construction de l'UE (2/3)

3. Contenu du cours :

- a) Introduction (éléments de base TI, contexte, images numériques, histogrammes, transformations simples)
- b) Modèles fondamentaux en TI : modèles linéaires (convolution), fréquentiels (Fourier ...), statistiques (histogramme, entropie ...), différentiels (gradient, EDP ...), ensemblistes (morphologie mathématique ...), discrets (maillage, connexité, distance ...)
- c) Etude des méthodes de TI du point de vue :
 - i. Prétraitement (restauration d'images, amélioration d'images, compression d'images)
 - ii. Traitement de bas-niveau (segmentation : approche contour, approche région)
 - iii. Traitement de haut-niveau (recherche de motifs, classification données / attributs de forme / couleurs / textures)

Une attention particulière sera accordée à la **compréhension fondamentale des méthodes abordées**, ainsi qu'à une **instanciation à des projets**, à l'état de l'art.

Les TPs (4 séances) seront à effectuer en Python (préféré) ou en Matlab.

Le Projet consiste en l'étude (présentation orale et code Python ou Matlab) d'une méthode de traitement de l'image à l'état de l'art (article) ou participation à l'analyse des données d'un benchmark / challenge (+2p)). Il nécessitera un **travail personnel** tout au long de l'UE.

3



3

Traitement d'images : construction de l'UE (3/3)

4. Prérequis

Des connaissances de base en **formation et représentation numérique des images** (capteurs, espaces de couleurs, histogrammes), ainsi que leur **traitement et analyse** (filtrage, extraction de contours, morphologie mathématique), sont nécessaires pour suivre cet enseignement.

Un bref rappel des prérequis sera effectué en début de cours.

5. Modalités de contrôle des Connaissances

QCM n°1 sur rappel des prérequis (20%) + QCM n°2 (40%) + TP/Projet (40%)

- Le 1^{er} QCM présentiel sera fait en fin de TP 2 (2^{ème} séance);
- TP/Projet - note sur la soutenance (fin du semestre) et le contenu du projet, ainsi que sur l'assiduité en TP (présence des CRs des TPs, qualitativement consistants)
- Le 2^{ème} QCM sera effectué en semaine d'examen.

4



4

Projets

Objectifs : étude d'une méthode de traitement de l'image à l'état de l'art

Le travail est à réaliser seul ou en binôme.

Les soutenances auront lieu en semaine d'examen (**semaine du 19 avril** - date à préciser).

Soumission des livrables : **8 avril 2022**

La soutenance durera **10 minutes en total**, par binôme/monôme.

Démo de la méthode proposée (de préférence en Python ou Matlab).

Deux types de sujets de projet sont possibles :

A. Challenge en imagerie biomédicale, robotique, spatial, véhicules autonomes etc. (+2p)

Sources potentielles :

- <https://grand-challenge.org/>
- <http://www.miccai.org/special-interest-groups/challenges/miccai-registered-challenges/>
- <https://www.kaggle.com/datasets>
- <https://www.kaggle.com/competitions>

B. Publication récente du domaine du traitement de l'image, à fort impact (nombre élevé de citations) faisant partie de la liste proposée (sur Moodle) ou proposée par vous (et validée par le responsable de l'UE).

Date limite du choix du sujet : **31 janvier 2022**

5



Traitement d'images
Image Processing

Plan du cours *Overall outline*



6

Plan general / Overall outline

- I. MODELES FONDAMENTAUX DE TRAITEMENT D'IMAGES**
- II. METHODES DE PRETRAITEMENT / PREPROCESSING**
- III. MÉTHODES DE TRAITEMENT DE BAS NIVEAU / LOW-LEVEL PROCESSING**
- IV. MÉTHODES DE TRAITEMENT DE HAUT NIVEAU / HIGH-LEVEL PROCESSING**

7



7

Plan general / Overall outline

- I. MODELES FONDAMENTAUX DE TRAITEMENT D'IMAGES**
 1. Modèle linéaire (convolution)
 2. Modèle fréquentiel (Fourier ...)
 3. Modèle statistique (histogramme, entropie ...)
 4. Modèle différentiel (gradient, équations à dérivées partielles - EDP ...)
 5. Modèle ensembliste (morphologie ...)
 6. Modèle discret (maillage, connexité, distance ...)
- II. METHODES DE PRETRAITEMENT / PREPROCESSING**
- III. MÉTHODES DE TRAITEMENT DE BAS NIVEAU / LOW-LEVEL PROCESSING**
- IV. MÉTHODES DE TRAITEMENT DE HAUT NIVEAU / HIGH-LEVEL PROCESSING**

8



8

Plan general / Overall outline

I. MODELES FONDAMENTAUX DE TRAITEMENT D'IMAGES

1. Modèle linéaire (convolution)
2. Modèle fréquentiel (Fourier ...)
3. Modèle statistique (histogramme, entropie ...)
4. Modèle différentiel (gradient, équations à dérivées partielles - EDP ...)
5. Modèle ensembliste (morphologie ...)
6. Modèle discret (maillage, connexité, distance ...)

II. METHODES DE PRETRAITEMENT / PREPROCESSING

1. Restauration d'images
2. Amélioration d'images / *Image Enhancement*
3. Compression d'images

III. MÉTHODES DE TRAITEMENT DE BAS NIVEAU / LOW-LEVEL PROCESSING

IV. MÉTHODES DE TRAITEMENT DE HAUT NIVEAU / HIGH-LEVEL PROCESSING



9

Plan general / Overall outline

I. MODELES FONDAMENTAUX DE TRAITEMENT D'IMAGES

1. Modèle linéaire (convolution)
2. Modèle fréquentiel (Fourier ...)
3. Modèle statistique (histogramme, entropie ...)
4. Modèle différentiel (gradient, équations à dérivées partielles - EDP ...)
5. Modèle ensembliste (morphologie ...)
6. Modèle discret (maillage, connexité, distance ...)

II. METHODES DE PRETRAITEMENT / PREPROCESSING

1. Restauration d'images
2. Amélioration d'images / *Image Enhancement*
3. Compression d'images

III. MÉTHODES DE TRAITEMENT DE BAS NIVEAU / LOW-LEVEL PROCESSING

1. Approche contour (frontière)
2. Approche région

IV. MÉTHODES DE TRAITEMENT DE HAUT NIVEAU / HIGH-LEVEL PROCESSING



10

10

Plan general / Overall outline

I. MODELES FONDAMENTAUX DE TRAITEMENT D'IMAGES

1. Modèle linéaire (convolution)
2. Modèle fréquentiel (Fourier ...)
3. Modèle statistique (histogramme, entropie ...)
4. Modèle différentiel (gradient, équations à dérivées partielles - EDP ...)
5. Modèle ensembliste (morphologie ...)
6. Modèle discret (maillage, connexité, distance ...)

II. METHODES DE PRETRAITEMENT / PREPROCESSING

1. Restauration d'images
2. Amélioration d'images / *Image Enhancement*
3. Compression d'images

III. MÉTHODES DE TRAITEMENT DE BAS NIVEAU / LOW-LEVEL PROCESSING

1. Approche contour (frontière)
2. Approche région

IV. MÉTHODES DE TRAITEMENT DE HAUT NIVEAU / HIGH-LEVEL PROCESSING

1. Recherche de motifs
2. Classification de données
 - i. Attributs de formes / *shapes*
 - ii. Attributs de couleur
 - iii. Attributs de texture

11



11

Bibliographie

- R.C. Gonzalez et R.E. Woods « *Digital Image Processing* », 2^{ème} ed., Addison Wesley 2002.
- J.P. Cocquerez et S. Philipp « *Analyse d'images : filtrage et segmentation* » Masson 1995.
- A. Rosenfeld et A.C. Kak « *Digital picture processing* » Academic Press London 1982.
- H. Maître (ss la direction de) « *Le traitement des images* » Hermès Lavoisier IC2 2003.
- J.R. Parker « *Algorithms for Image Processing and Computer Vision* » Wiley & Sons 1997.
- S. Bres, J.M. Jolion, F. Lebourgeois « *Traitements et analyse des images numériques* » Hermès Lavoisier 2003
- I.T. Young, J.J. Gerbrands et L.J. Van Vliet « *Fundamentals of Image Processing* » Université de Delft. (sur internet : <http://www.ph.tn.tudelft.nl/~lucas/publications/1995/FIP95TYJGLV/FIP2.2.pdf>)
- D. Lingrand « *Introduction au Traitement d'images* » Vuibert 2004

12



12

Partie I

Modèles fondamentaux en TI

Fundamental models in image processing



13

Plan de la première partie

1. INTRODUCTION

- Intérêt du traitement d'image
- Applications du traitement d'images
- Vocabulaire de base de l'image et du traitement de l'image

2. MODÈLES FONDAMENTAUX

- a. Modèle statistique (histogramme, entropie ...)
- b. Modèle linéaire (convolution)
- c. Modèle fréquentiel (Fourier ...)
- d. Modèle différentiel (gradient, EDP ...)
- e. Modèle ensembliste (morphologie ...)
- f. Modèle discret (maillage, connexité, distance ...)

14



14

Plan de la première partie

1. INTRODUCTION

Intérêt du traitement d'image

Applications du traitement d'images

Vocabulaire de base de l'image et du traitement de l'image

2. MODÈLES FONDAMENTAUX

- a. Modèle statistique (histogramme, entropie ...)
- b. Modèle linéaire (convolution)
- c. Modèle fréquentiel (Fourier ...)
- d. Modèle différentiel (gradient, EDP ...)
- e. Modèle ensembliste (morphologie ...)
- f. Modèle discret (maillage, connexité, distance ...)

15



15

Domaines d'application

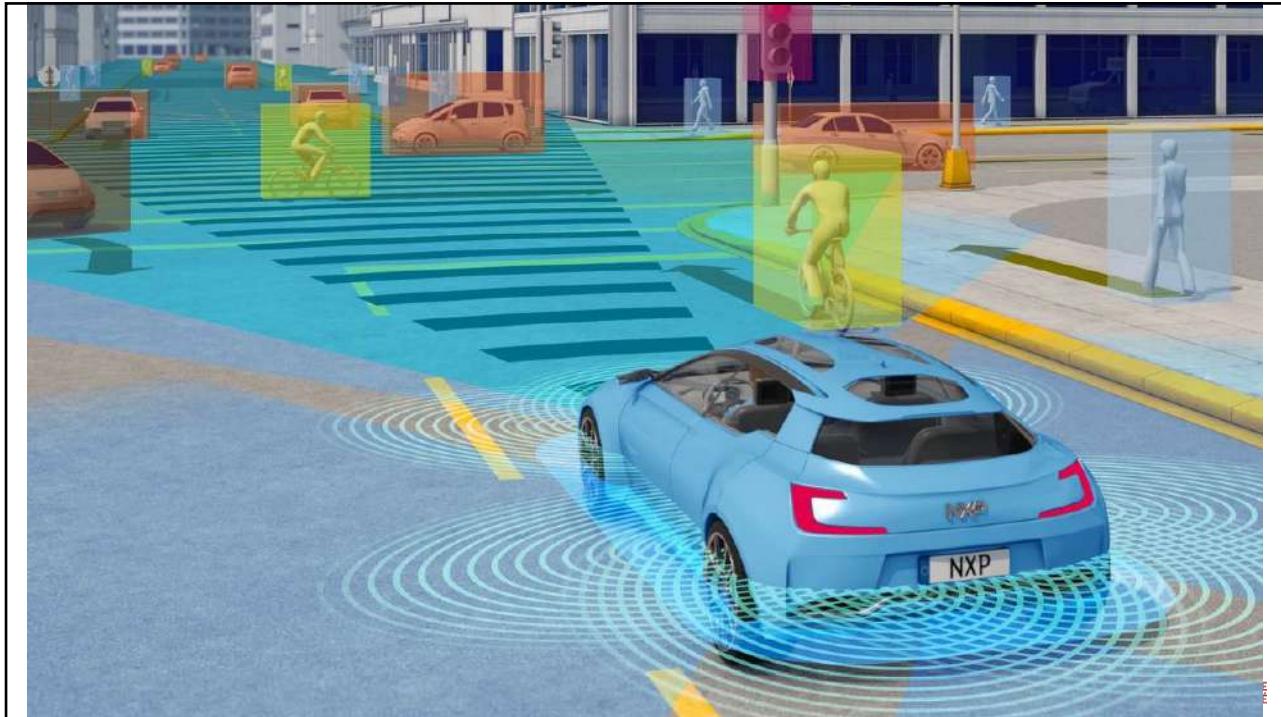
- Aide à la conduite automobile
- ...



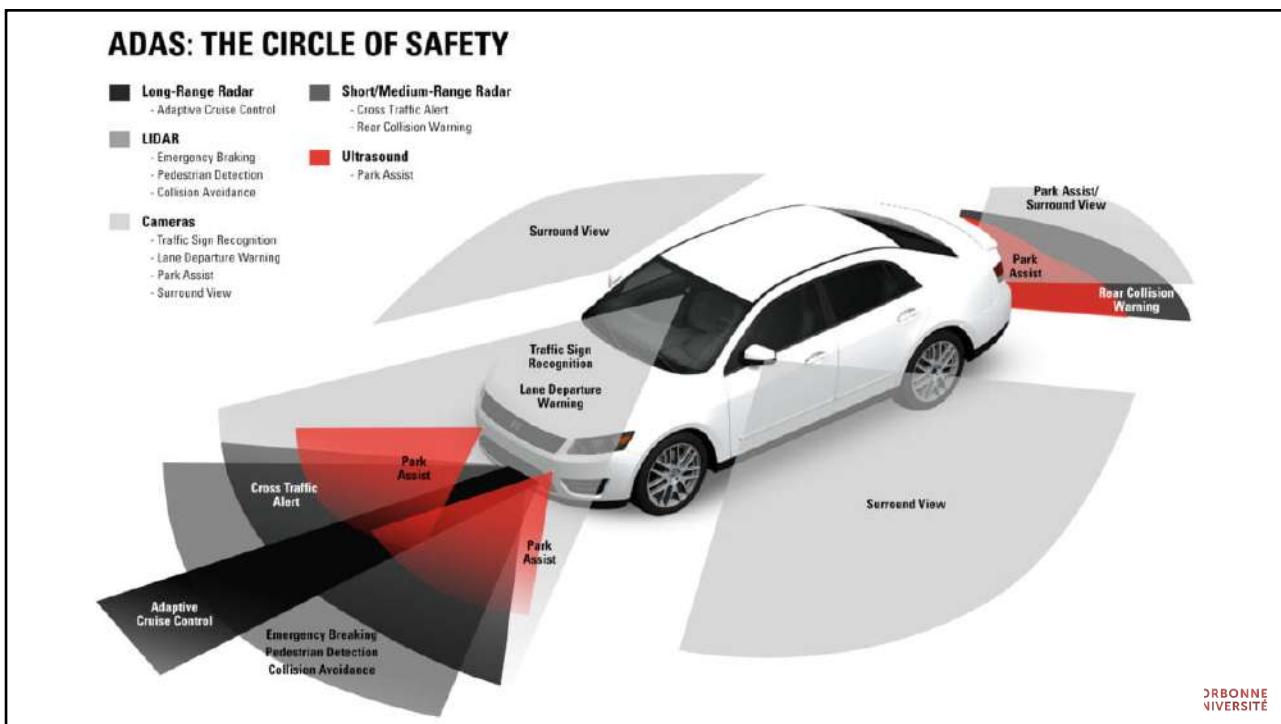
16



16



17



18

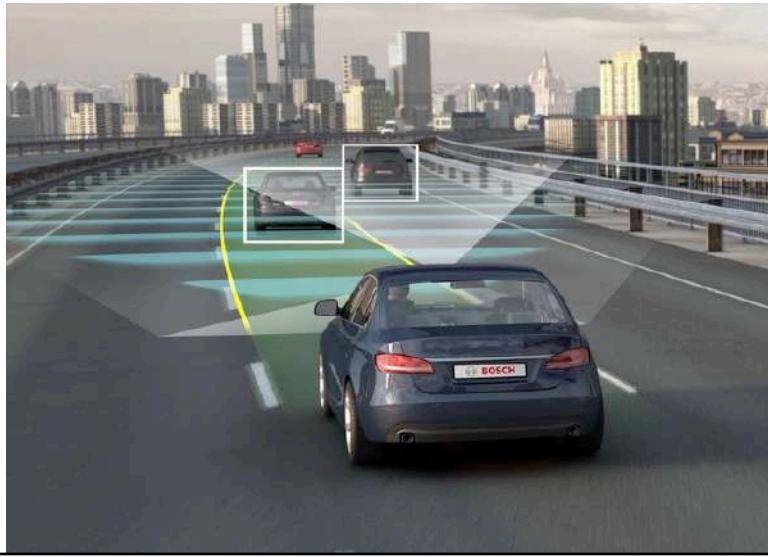
Niveau 1 : mise en place des premiers éléments

prémisses des aides à la conduite.

voiture capable de **gérer la partie longitudinale** de la conduite (i.e. régulateur de vitesse adaptatif - gère automatiquement l'accélérateur et le frein), tandis que le conducteur se charge du contrôle latéral de l'auto, au volant.

aides à la conduite : **alertes au conducteur**, voire **action automatique** (détection angle mort, alerte franchissement file, alerte risque collision, freinage d'urgence automatique, park assist, etc.).

Avancement : mise en place progressive depuis les années 90 - berlines haut de gamme, notamment premium (i.e. régulateur de vitesse adaptatif Distronic : [Mercedes Classe S](#))



19

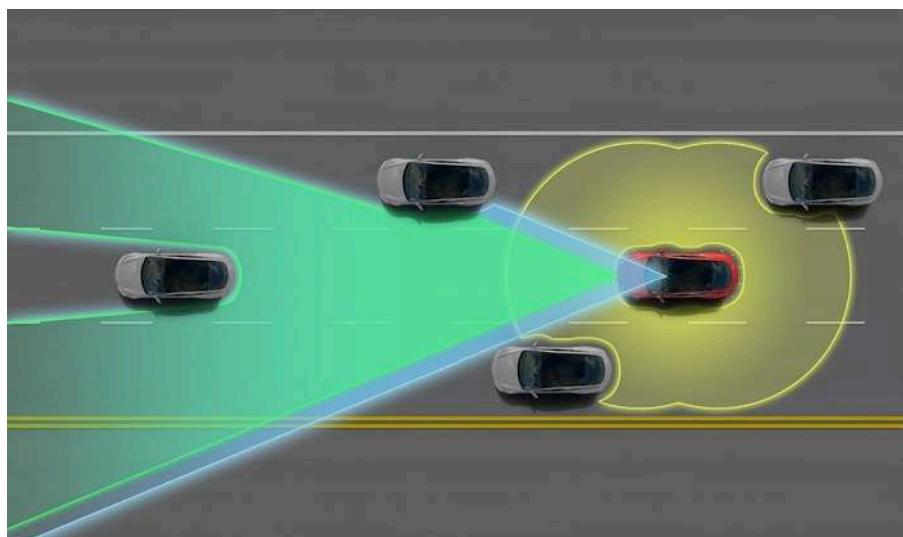
Niveau 2 : la co-conduite

Niveau maximal déjà atteint par certains véhicules du marché d'aujourd'hui.

Combiner fonctions du niveau 1 avec le **contrôle longitudinal et latéral automatisé** (volant corrige la trajectoire pour rester inscrit dans sa file de circulation), mais avec une présence active du conducteur, sous surveillance permanente.

Le système contrôle l'acuité du conducteur et agit en cas de non-réaction.

Avancement : milieu des années 2000, (direction active sur Lexus LS 460 en 2006).



20

Niveau 3 : le début du lâcher prise

Le conducteur laisse le véhicule entièrement assurer certaines phases de conduite, notamment sur autoroute, dans les bouchons et dans les parkings. Il doit cependant être prêt à reprendre le volant à tout moment, dès que le système le lui suggère.

Audi évoque cela dans sa toute dernière A8., dans les embouteillages jusqu'à 60 km/h, tandis que Tesla possède aussi ce type de capacité de niveau 3 (bouchons, conduite sur autoroute, parking...).

Ces fonctions ne sont pas encore activées car la réglementation actuelle n'autorise pas encore ces phases de conduite.

Niveau d'avancement : sous réserve d'évolution de la réglementation.



21

Niveau 4 : l'émancipation

certaines phases de conduite (autoroute, bouchons, parking) sont prises en charge par l'auto + le conducteur peut vaquer à d'autres occupations

(il peut très bien faire une sieste)

redondance des multiples capteurs embarqués et sophistication poussée des fonctionnalités autonomes + évolution réglementation

le conducteur peut choisir de reprendre le volant.

22



22

Niveau 5 : L'ère des robots

Autonomie totale

Ère véhicules entièrement automatiques, sans volant, comme l'illustraient les fameuses petites Google cars.

Navettes automatiques et taxis partagés

Défi technologique complet. Il faudra vraisemblablement compter au moins une quinzaine d'années ...



23

Domaines d'application

- **Aide à la conduite automobile**

Organisation Internationale des Constructeurs Automobiles (OICA)
barème européen avec 6 niveaux d'autonomie pour les véhicules :

- Niveau 0** : aucune autonomie (mécanismes de type radar de recul etc.)
- Niveau 1** : confier à l'intelligence une tâche de conduite (régulateur de vitesse adaptatif)
- Niveau 2** : tâche complète est confiée au véhicule (fonction Park Assist)
- Niveau 3** : déléguer une partie de sa conduite (fonctionnalité Assistant Embouteillage)
- Niveau 4** : il s'agit d'une autonomie quasi complète (i.e. "City Park Full Automatic")
- Niveau 5** : c'est la voiture 100% autonome (i.e. Google Car)

24

Première course automobile sans chauffeurs : une voiture autonome peut en doubler une autre

Las Vegas
7 Janvier 2022
185 km/h



25

25

Domaines d'application

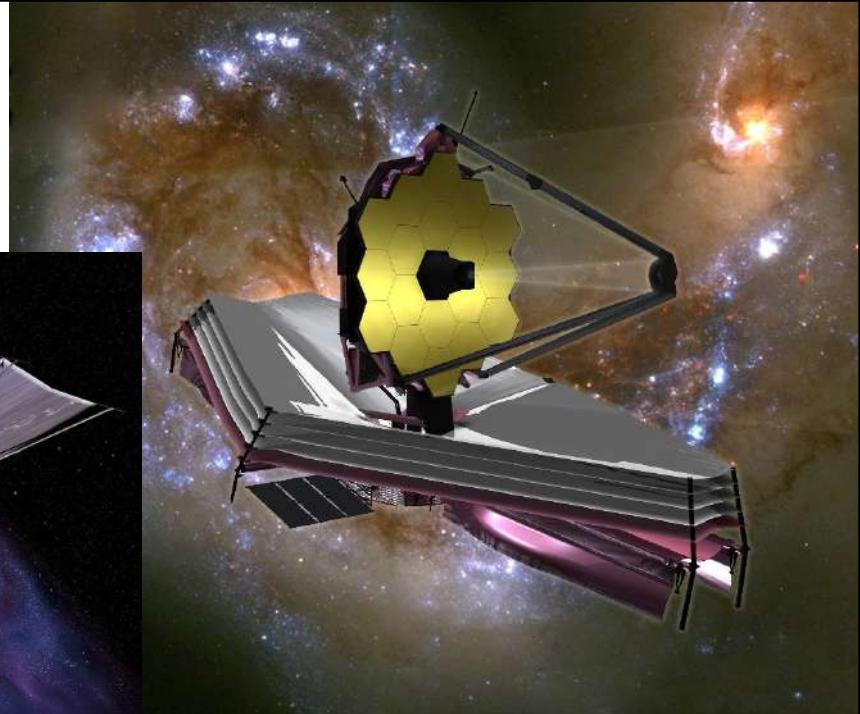
- Aide à la conduite automobile
- **Spatial**
- ...



26

26

- Télescope James Webb
- Lancement 25/12/2021
- de Guyane (Kourou) par Ariane 5
- miroir principal de 6,5 mètres de diamètre, formé de 18 miroirs plus petits.
- coût final 10 milliards dollars.



27

Domaines d'application

- Aide à la conduite automobile
- Spatial
- **Médical**
- ...

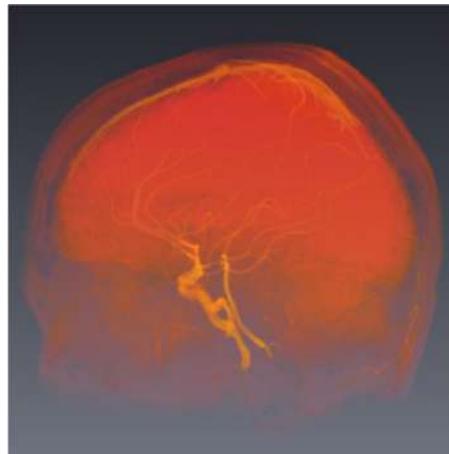


28

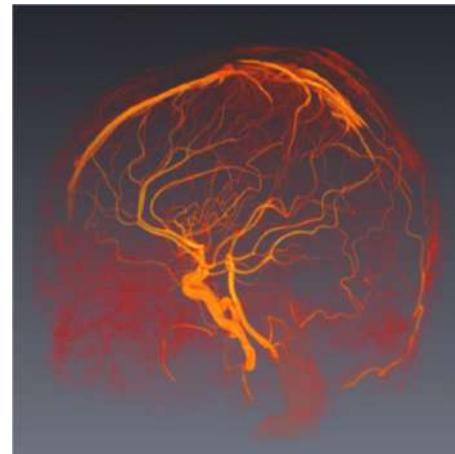
28

Domaines d'application

- Aide à la conduite automobile
- Spatial
- **Médical**
- ...

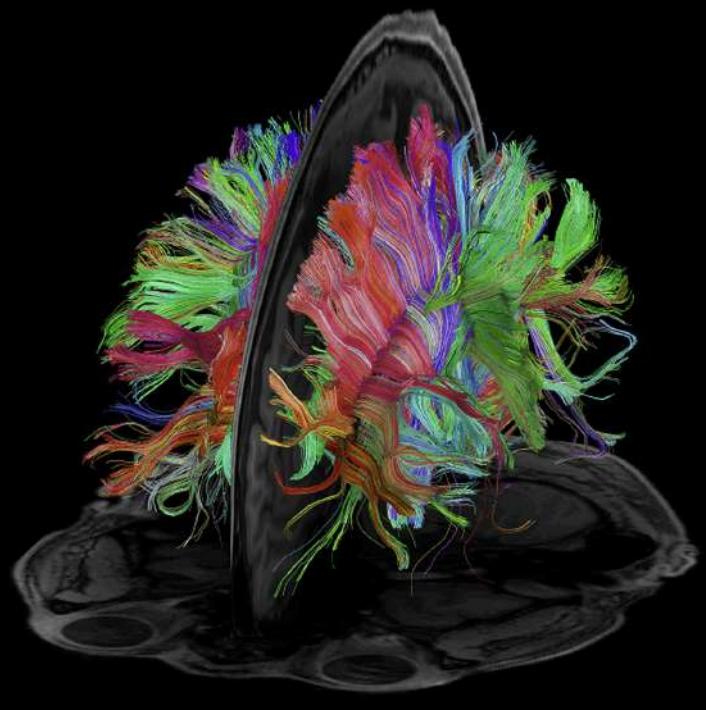
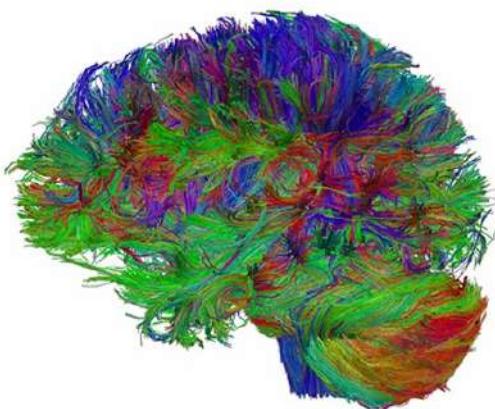


29

 SORBONNE
UNIVERSITÉ

29

- Tractographie
- Neurosciences - méthode utilisée pour mettre en évidence les voies neuronales
- Technique d'IRM de diffusion.



30

Domaines d'application

- Aide à la conduite automobile
- Spatial
- Médical
- **Construction automobile**
- ...

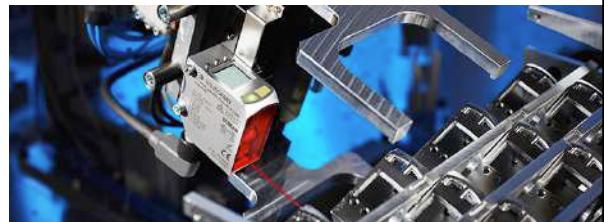
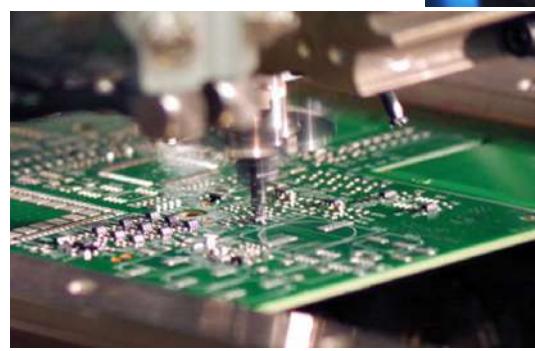


 SORBONNE
UNIVERSITÉ

31

Domaines d'application

- Aide à la conduite automobile
- Spatial
- Médical
- Construction automobile
- **Electronique**
- ...

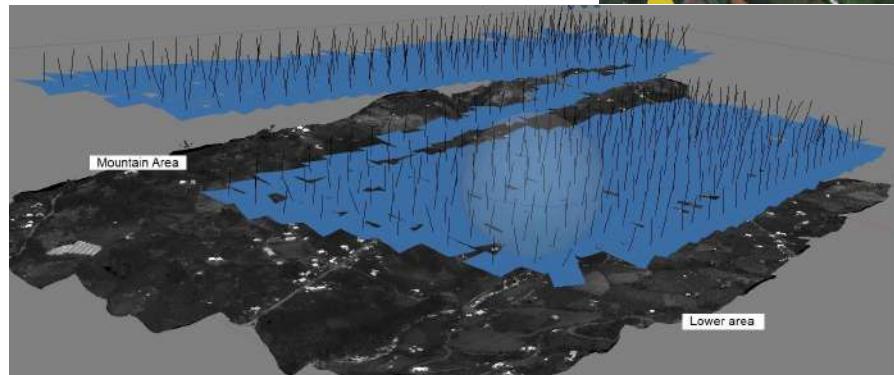


 SORBONNE
UNIVERSITÉ

32

Domaines d'application

- Aide à la conduite automobile
- Spatial
- Médical
- Construction automobile
- Electronique
- **Agriculture**
- ...



33

SORBONNE
UNIVERSITÉ

33

Domaines d'application

- Aide à la conduite automobile
- Spatial
- Médical
- Construction automobile
- Electronique
- Agriculture
- **Colorisations de film**
- ...

Annabelle -
Serpentine
Dance
(1895)
[Edison]



Portrait de Greta Garbo pour *La Courtesane*, 1931.
La même image colorisée

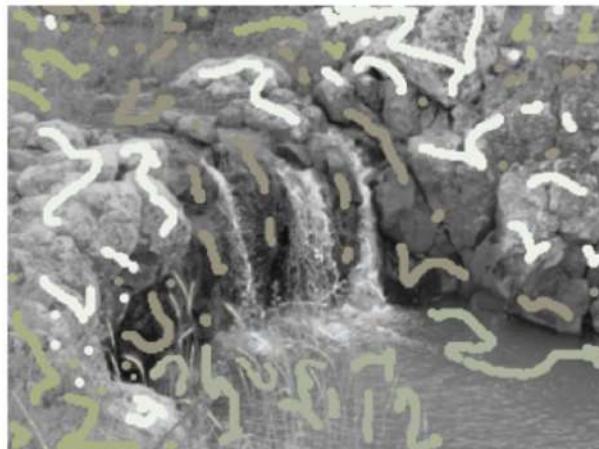
34

<http://www.youtube.com/watch?v=p94yQ8cvTHg&NR=1>

SORBONNE
UNIVERSITÉ

34

Processus de colorisation par diffusion. À gauche : traits de couleurs d'un utilisateur. À droite : résultat de colorisation.



35



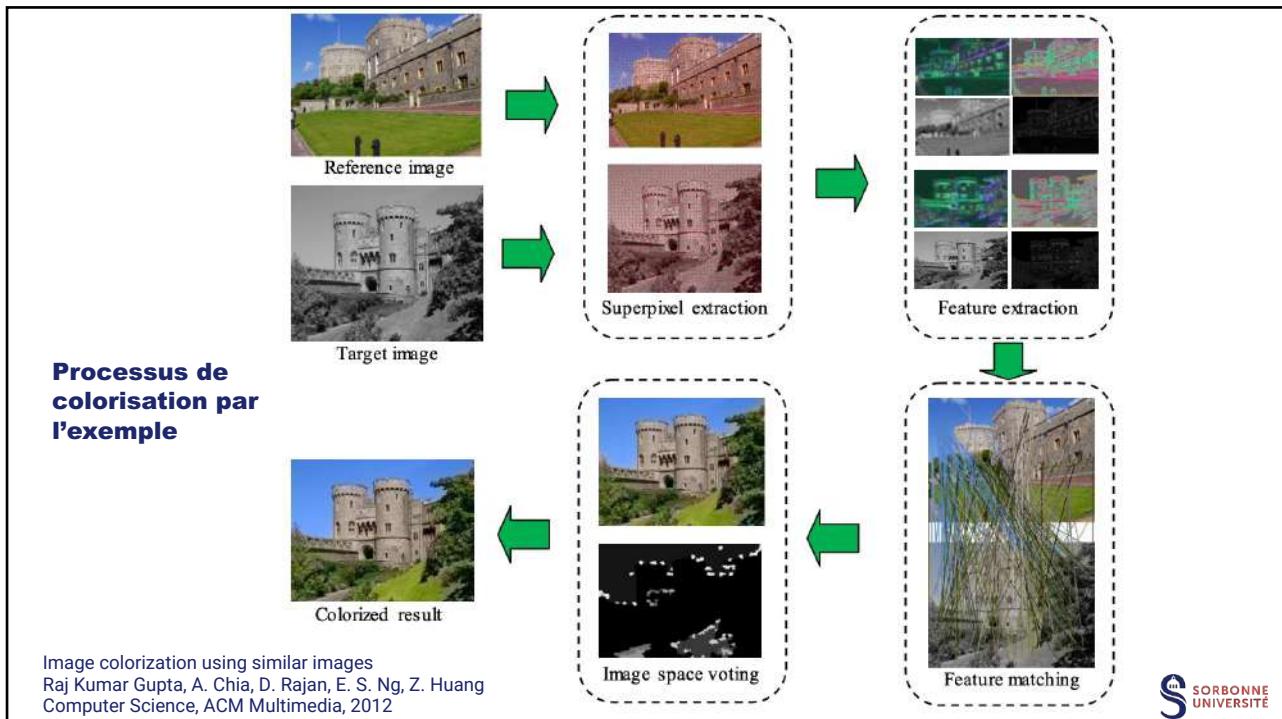
35

Processus de colorisation par l'exemple

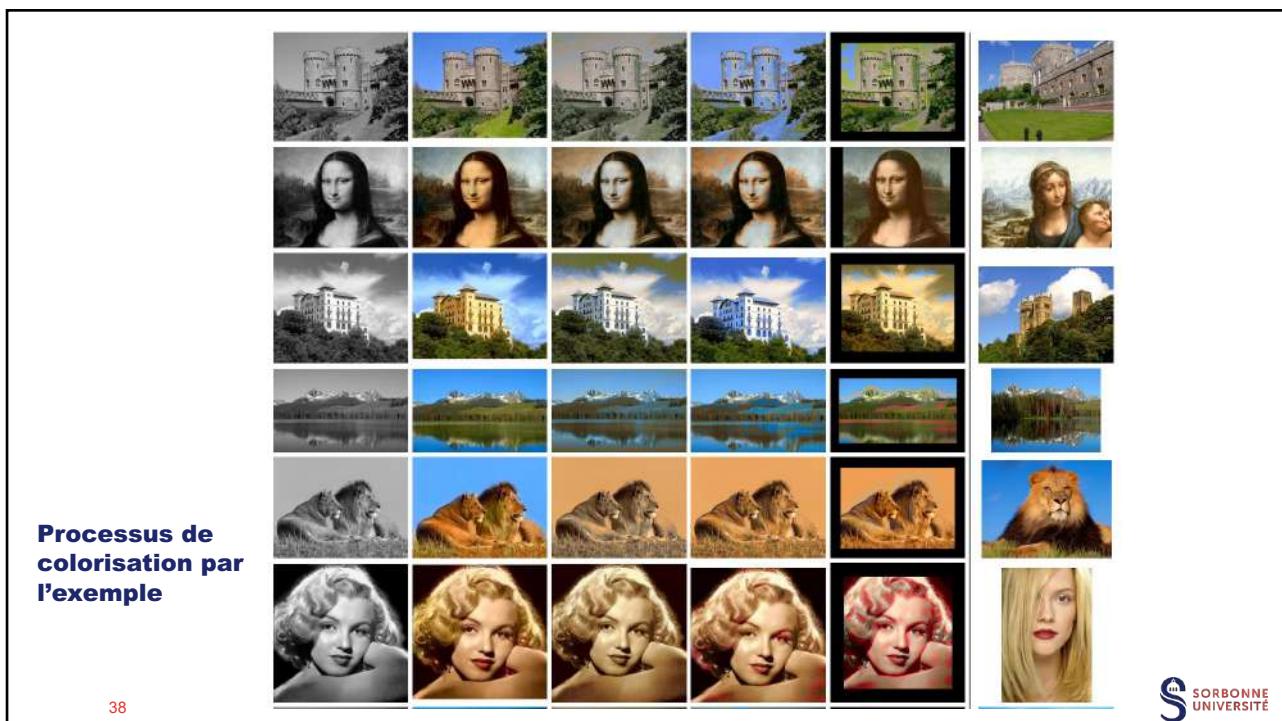
36



36



37



38

19

21 old films from 1895 to 1902 colorized and upscaled in 60 fps, with sound

39 https://www.youtube.com/watch?v=YZuP41ALx_Q SORBONNE UNIVERSITÉ

39

Domaines d'application

- Aide à la conduite automobile
- Spatial
- Médical
- Construction automobile
- Electronique
- Agriculture
- Colorisations de film
- **Restauration d'images**
- ...



40

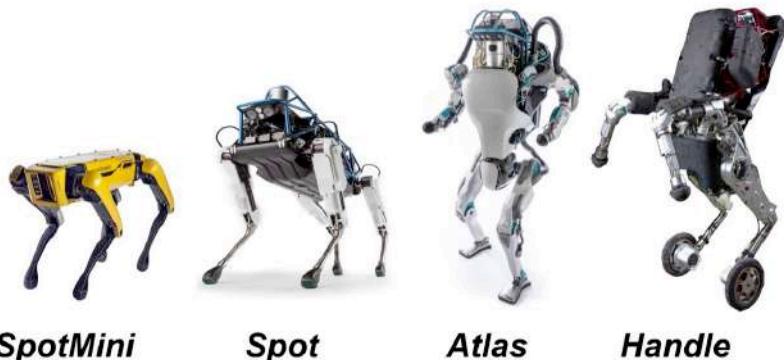
40

Domaines d'application

- Aide à la conduite automobile
- Spatial
- Médical
- Construction automobile
- Electronique
- Agriculture
- Colorisation de film
- Restauration d'images
- **Robotique**
- ...

Platforms

Boston Dynamics 



SpotMini

Spot

Atlas

Handle

41

https://www.youtube.com/watch?v=_sBBaNYex3E

 SORBONNE
UNIVERSITÉ

41

Domaines d'application

- ...
- Construction automobile
- Electronique
- Agriculture
- Colorisation de film
- Restauration d'images
- Robotique
- **Militaire**



 SORBONNE
UNIVERSITÉ

42

Domaines d'application

- ...
- Construction automobile
- Electronique
- Agriculture
- Colorisation de film
- Restauration d'images
- Robotique
- Militaire
- Sécurité



43



STEGANOGRAPHIE



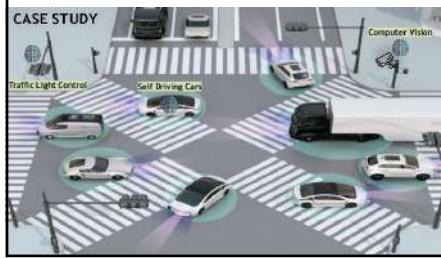
watermarking

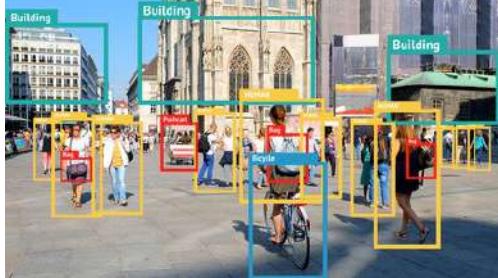


43

Domaines d'application

- Aide à la conduite automobile
- Spatial
- Médical
- Construction automobile
- Electronique
- Agriculture
- Colorisation de film
- Restauration d'images
- Robotique
- Militaire
- **Cité intelligente**





44

Exemples d'images numériques

Image 2D :

objet représenté par un tableau bidimensionnel de surfaces élémentaires (**pixels***)

Séquence vidéo (2D+t) :

scène dynamique présentant des objets 2D en mouvement. Les séquences vidéo 2D sont une juxtaposition d'images 2D, où le temps est vu comme une troisième dimension

Image volumique (3D) :

objet représenté par un tableau tridimensionnel de volumes élémentaires (**voxels****)

- Notion de profondeur z
- Un volume peut être vu comme un empilement d'images 2D (ex. des coupes scanner pour la reconstruction 3D)

Séquence volumique (3D+t) :

scène dynamique présentant des objets 3D en mouvement

45

*pixel = picture element (élément d'image)
**voxel = volumetric pixel or volume pixel (pixel volumétrique)



45

Exemples de traitement d'images

Amélioration : augmenter la qualité de la perception visuelle d'une image

Restauration : compenser les dégradations (bruit, flou, ...)

Compression : stocker et transférer efficacement

Segmentation : délimiter les "objets"

Reconstruction 3D : obtenir un volume à partir de plans (images)

Recalage : mise en correspondance d'images

Représentation : modéliser

Bas niveau : couleur, forme, régions, contours

Haut niveau : caractéristiques (features), motifs, texture, indices statistiques, graphes

Analyse : convertir en informations

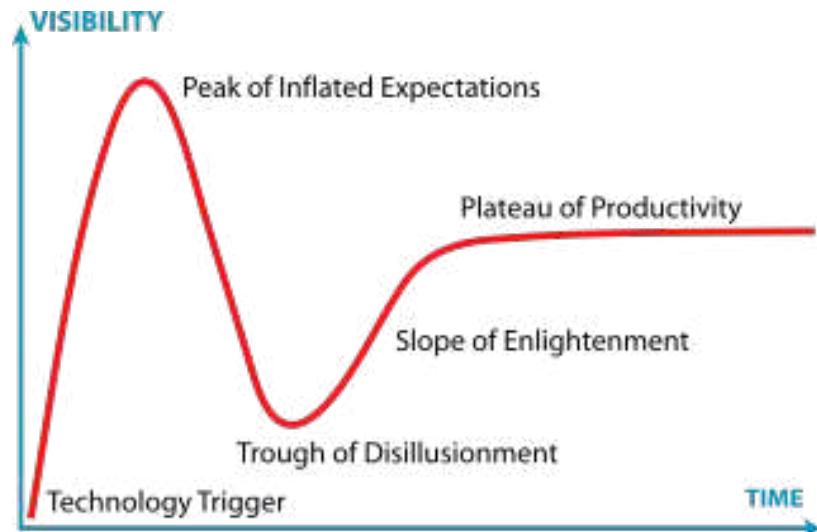
Reconnaissance / Compréhension : identifier / comprendre le contenu

46



46

Diagramme de Gartner



SORBONNE
UNIVERSITÉ

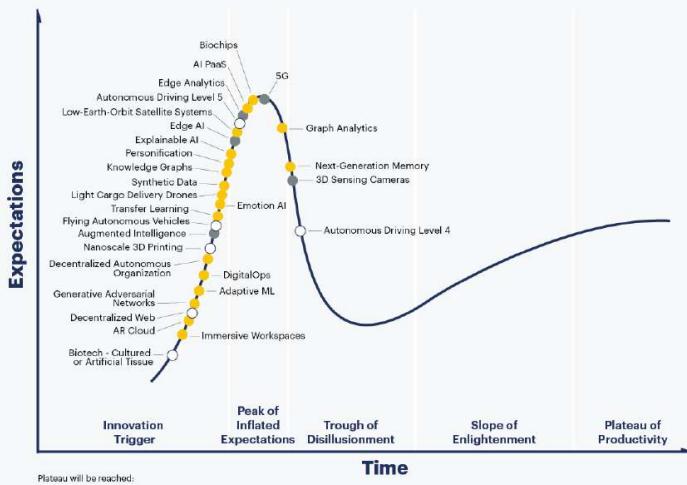
47

47

Diagramme de Gartner

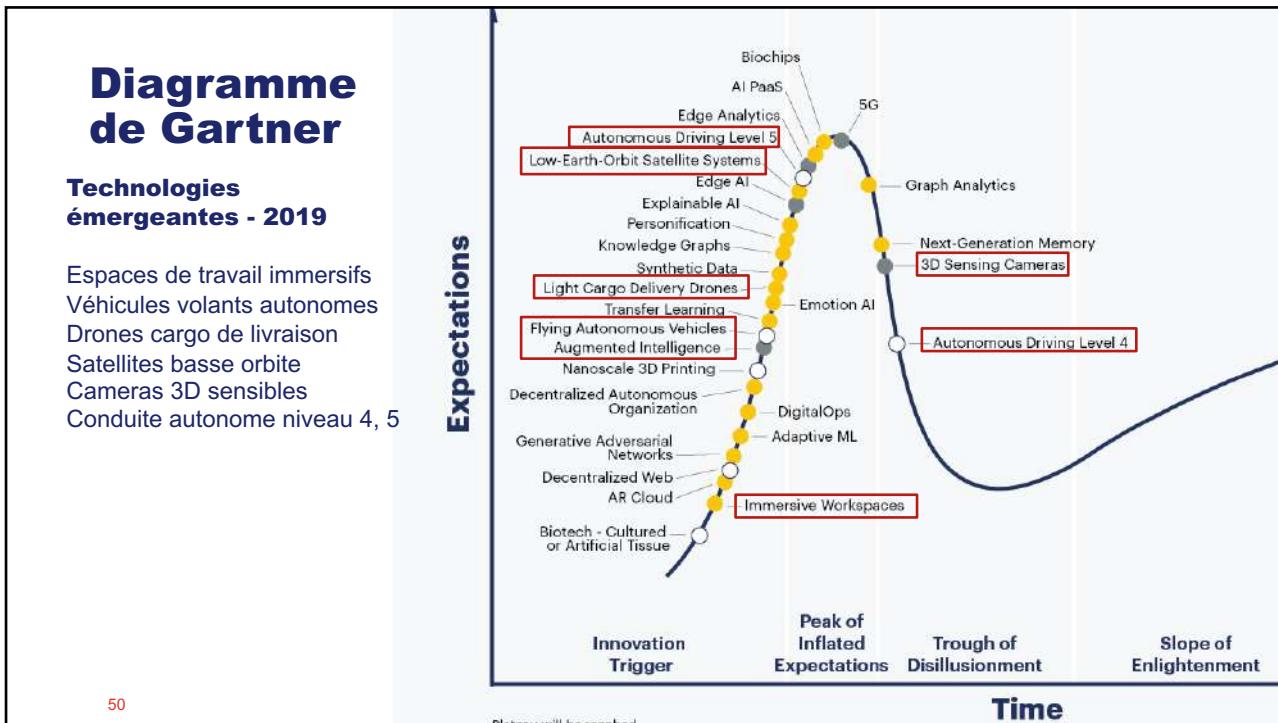
Technologies émergeantes - 2019

Gartner Hype Cycle for Emerging Technologies, 2019

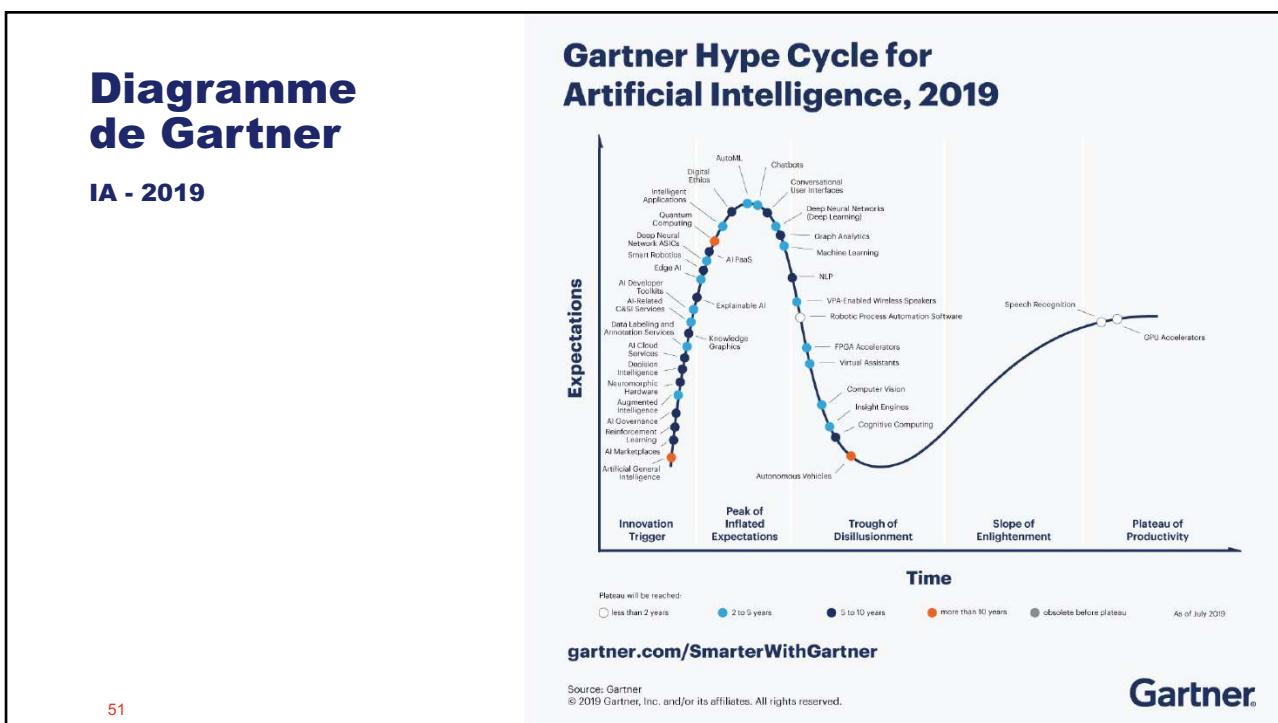


49

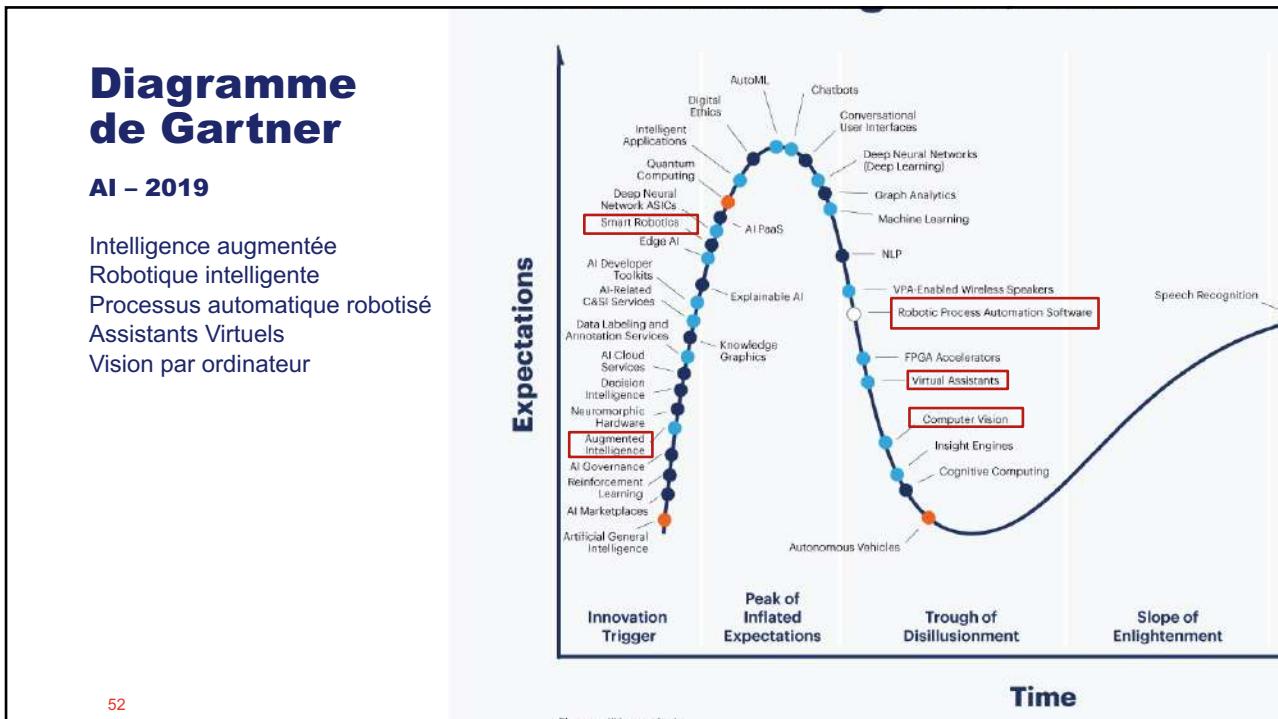
49



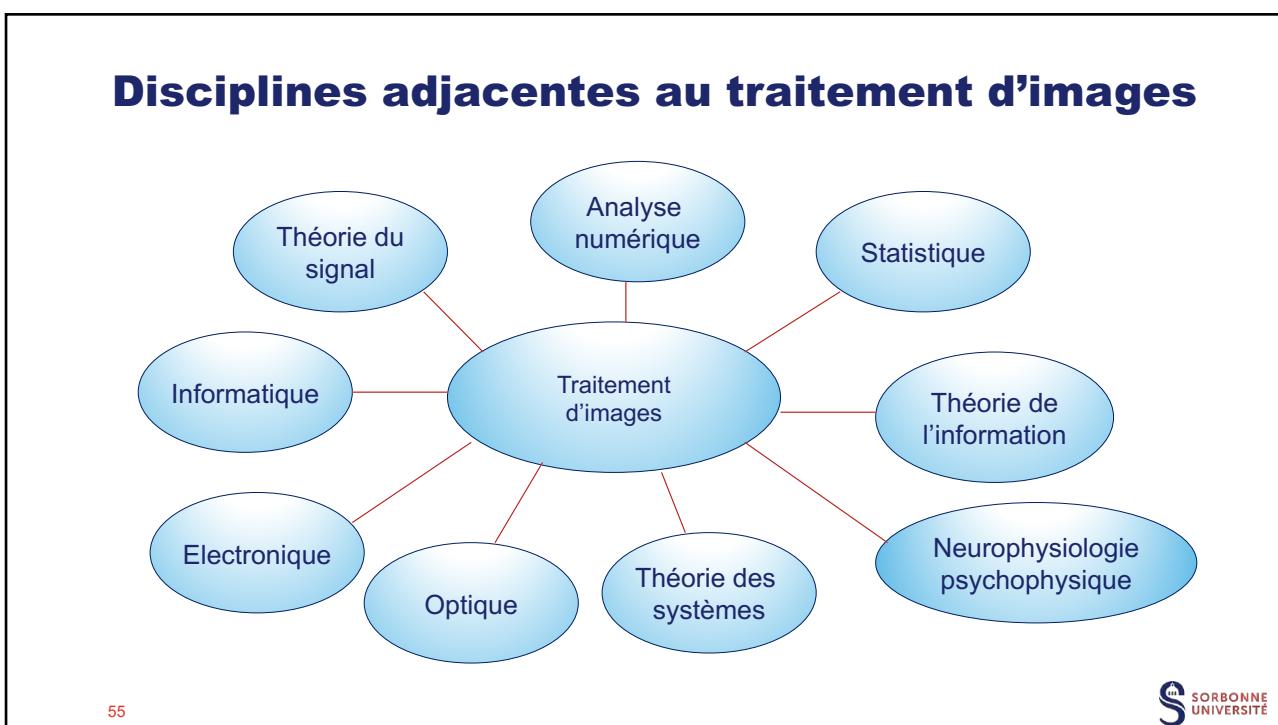
50



51

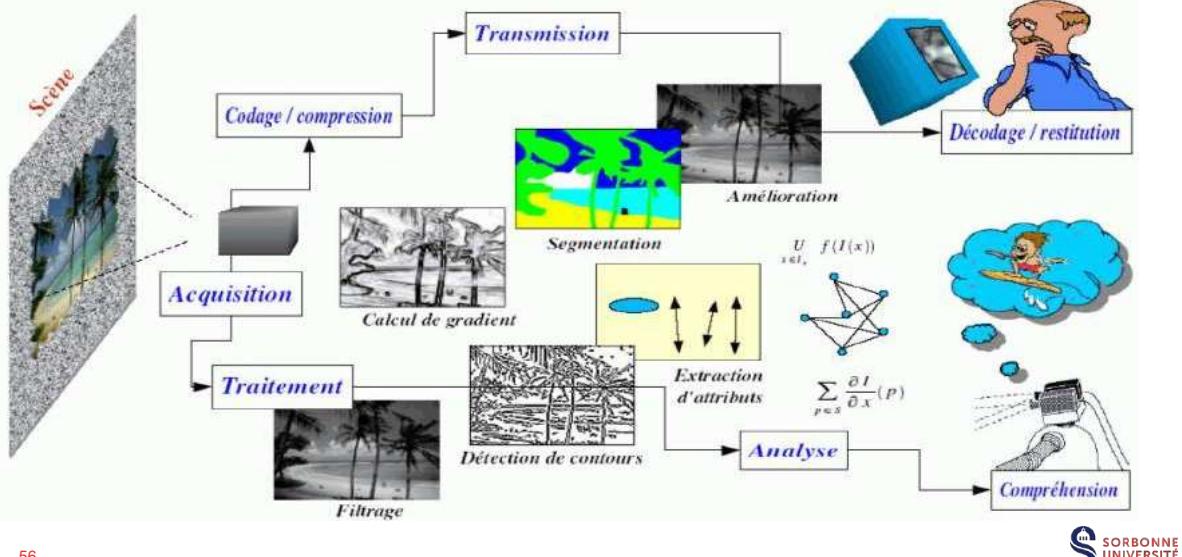


52



55

Vision globale

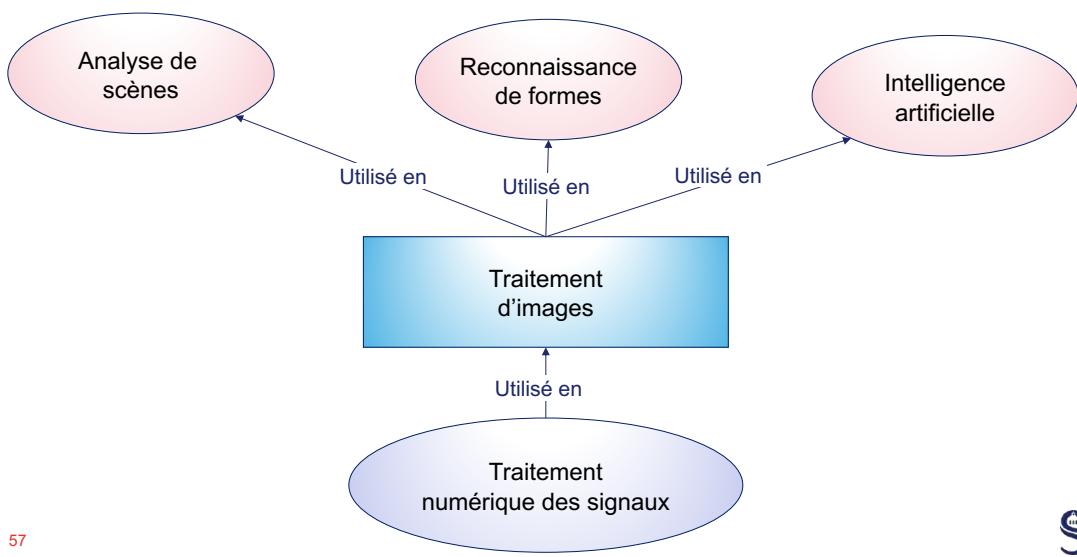


56

 SORBONNE
UNIVERSITÉ

56

Positionnement du traitement d'images (TI)

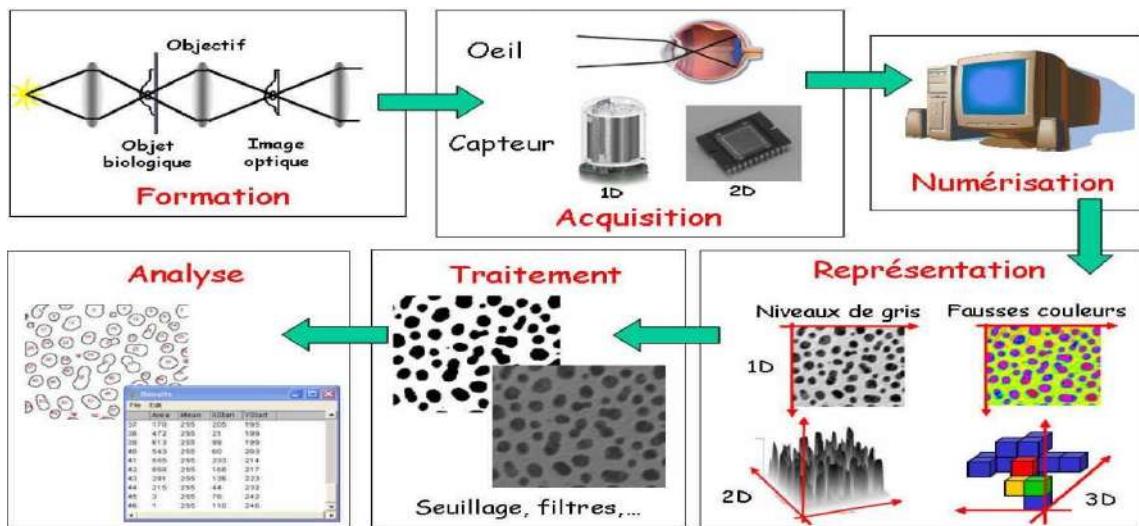


57

 SORBONNE
UNIVERSITÉ

57

Traitement et analyse d'images



S SORBOONE
UNIVERSITÉ

58

58

Manipulations d'images : typologies

Manipulations sur les images - 3 grandes catégories :

- **Composition** : assembler différentes images pour constituer une nouvelle image avec par exemple des effets de transparence ou de masquage ;
- **Traitement** : modification d'une image donnée soit par une transformation géométrique (i.e. rotation), soit pour modifier son contenu (i.e. élimination du bruit, renforcement des contours) ;
- **Analyse** : extractions d'informations structurelles ou sémantiques (contours, objets 3D)

NB : Ces trois catégories sont loin d'être indépendantes :

- la composition nécessite des techniques de traitement (i.e. pour changer la taille d'une image avant de la composer)
- l'analyse fait la plupart du temps appel à des techniques de traitement, notamment par filtrage.

59

Titre de la présentation

S SORBOONE
UNIVERSITÉ

59

Composition d'images

Art

60



60

Traitements d'images

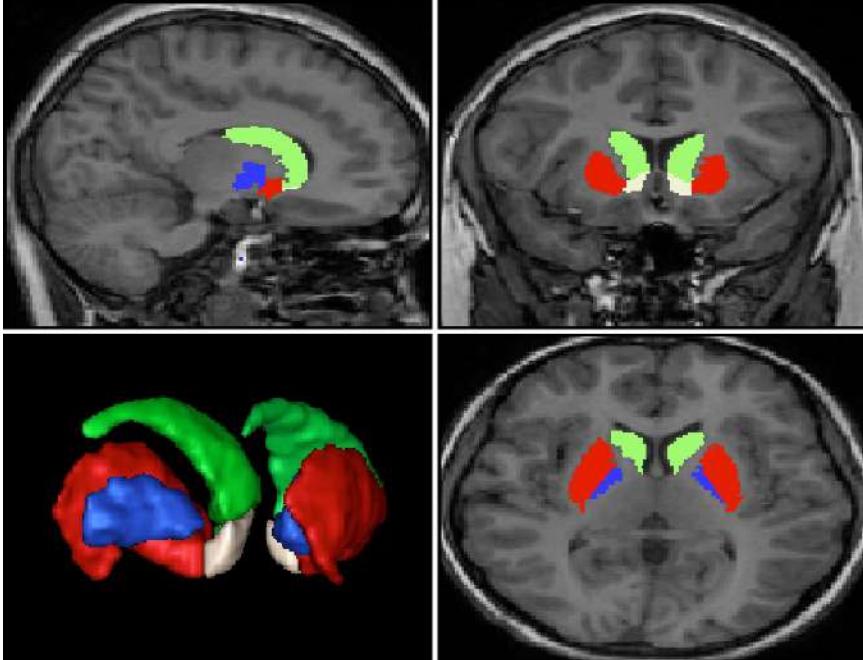
Télédétection (Remote sensing)



61

Traitement et Analyse d'Images

Applications médicales
Imagerie médicale
IRM cerveau
Hippocampe



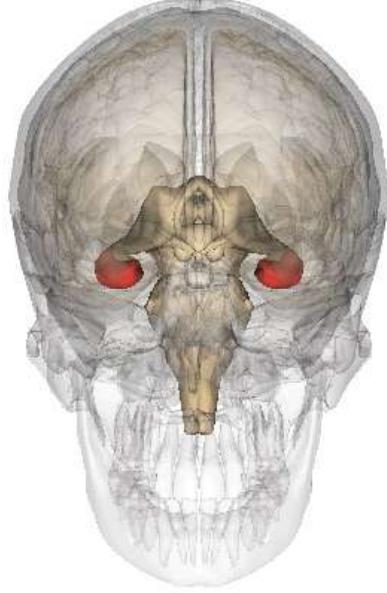
62



62

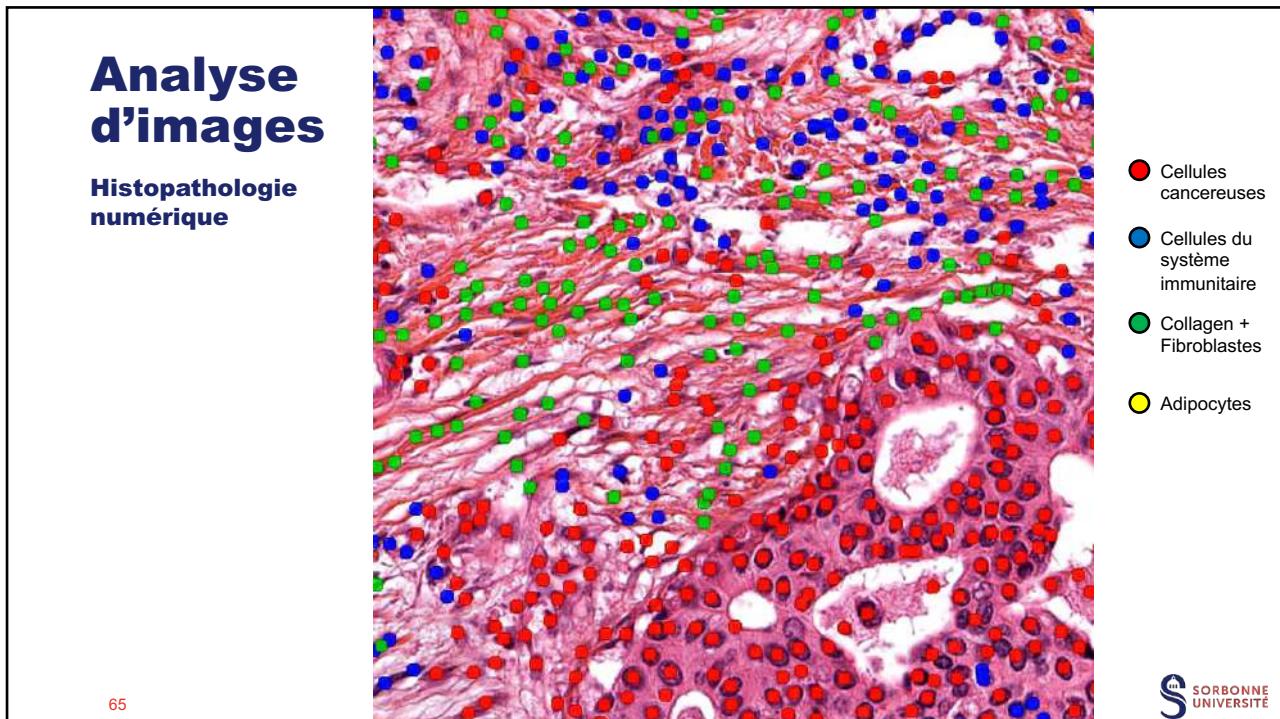
Traitement et Analyse d'Images

Applications médicales
Imagerie médicale
Hippocampe



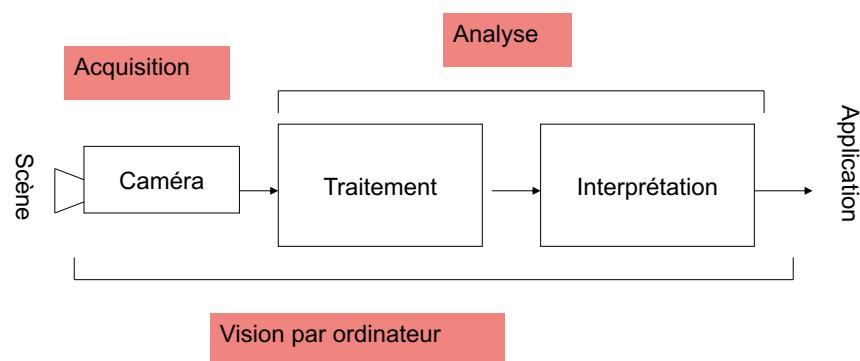
63 Images generated by Life Science Databases (LSDB).
URL : <https://commons.wikimedia.org/w/index.php?curid=7887142>

63



65

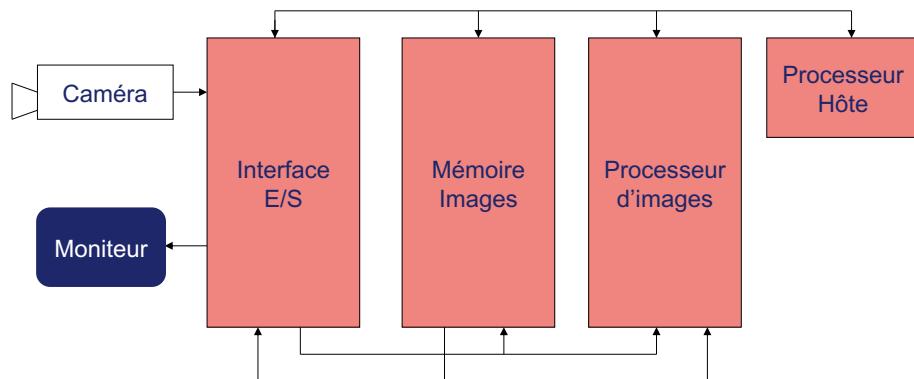
Constitution d'un système de vision : principe



66

66

Constitution d'un système de vision : matériel



67



67

Constitution d'un système de vision

1. Interface E/S

Numérisation des images
Visualisation N/B et Couleurs

2. Mémoire Images

Mémoire images source
Mémoire images traitées
- différents formats : 256x256 ; 512x512 ; 1024 x 1024 ; mode AOI (Automatic Optic Inspection)

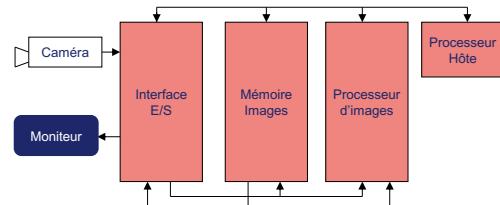
3. Processeur d'images

ALU (Arithmetic Logic Unit) ; Convolution ; Morphologie mathématique ; Histogramme ...

4. Processeur hôte

3 rôles :

- commande le processeur d'images
- archivage des images
- traitements complémentaires de ceux du processeur d'images



68



68

Etapes du traitement numérique d'une image

3 étapes fondamentales du TI :

- **Acquisition** : scène physique \Rightarrow représentation numérique.
- **Traitement** : extraction de l'information pertinente par segmentation \Rightarrow description structurelle de l'image.
- **Interprétation** : description structurelle \Rightarrow description sémantique.

69



69

Acquisition : Numérisation d'une image vidéo

1. Description d'un signal vidéo analogique :

- Normes :
 - CCIR (Europe) 625 lignes 1 trame 40 ms (50 Hz)
 - RS 170/33 (US) 525 lignes 1 trame 33 ms (60 Hz)

2. Echantillonnage spatial :

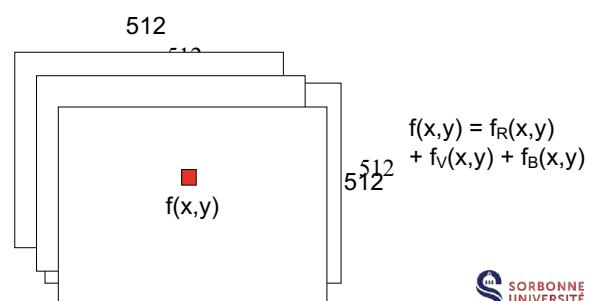
Découpage du signal analogique \Rightarrow pixels
 ex : caméra CCD standard : échantillonnage par construction ($512 \times 512 = 262144$ pixels) nécessite de composant électronique rapide (convertisseur analogique numérique flash)

3. Quantification : CAN \Rightarrow niveaux de gris

ex : codage sur 8 bits \Rightarrow 256 niveaux ($f(x,y)$)

\Rightarrow cas de la couleur : 3 plans (RVB ou ITS)

$$\begin{aligned} f(x,y) &= n \\ 0 &\leq n \leq 255 \end{aligned}$$



71

71

Définition d'une image numérique

Matrice dont la valeur de chaque élément représente une intensité discrète de la lumière

Plan discret dérivé d'une image analogique après numérisation (**digitization**)

- Echantillonnage spatial (**sampling**) : discréétisation des coordonnées de l'image réelle
- Quantification des luminances (**quantization**) : discréétisation des intensités de l'image réelle

Echantillonnage spatial (sampling)

- **Définition** (spatiale) de l'image
 - » Pas de division du plan image : nombre d'éléments par unité de longueur
 - » Plus petits détails discernables dans l'image

NB : une résolution spatiale trop faible provoque des effets de "crenelage" (aliasing)

Quantification des luminances (quantization)

- L'intensité I est quantifiée sur m bits et peut prendre $L = 2^m$ valeurs : $I \in [0, \dots, 2^m - 1]$
- Plus petit changement d'intensité discernable dans l'image

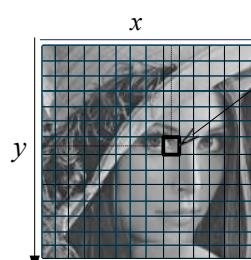
NB : Un quantification trop faible provoque des "faux contours"

72



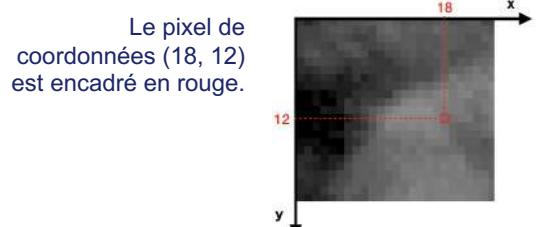
72

Images numériques



L'**échantillonnage** est le procédé de discréétisation spatiale d'une image consistant à associer à chaque zone rectangulaire $R(x,y)$ d'une image continue une unique valeur $I(x,y)$.

On parle de **souséchantillonnage** lorsque l'image est déjà discréétisée et qu'on diminue le nombre d'échantillons.



La **quantification** désigne la limitation du nombre de valeurs différentes que peut prendre $I(x,y)$.



Une **image numérique** est une image **échantillonnée** et **quantifiée**.

73



73

34

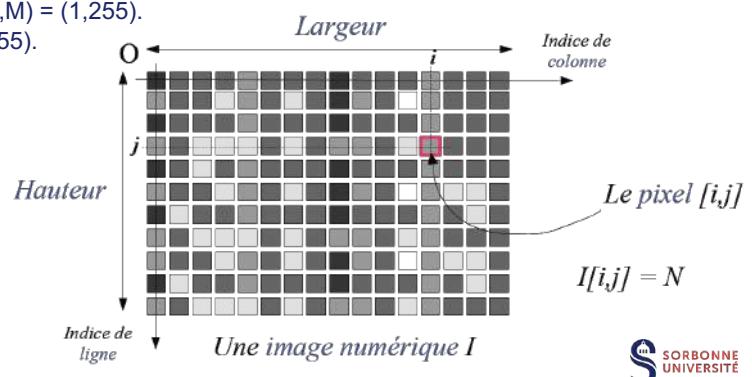
Définition d'une image numérique

$$I : [0, L-1] \times [0, C-1] \rightarrow [0, M]^p$$

Définit une image de **L** lignes et **C** colonnes

- Si I est une image binaire, $(p, M) = (1, 1)$.
- Si I est une image en niveau de gris, $(p, M) = (1, 255)$.
- Si I est une image couleur, $(p, M) = (3, 255)$.

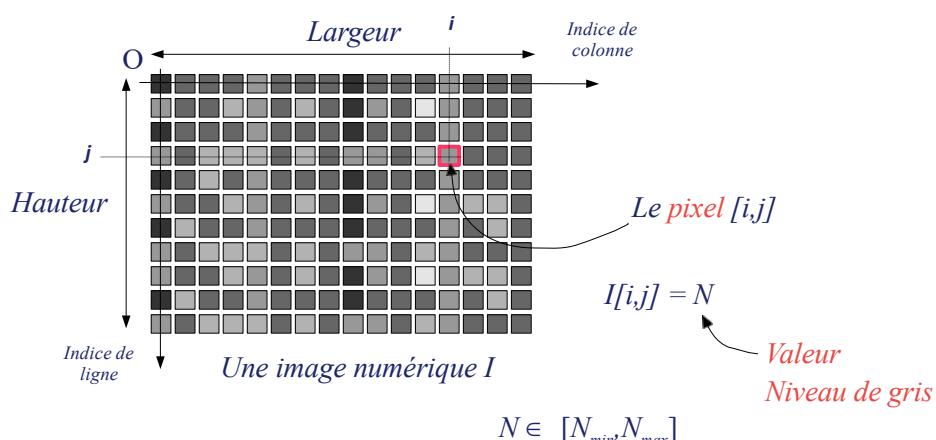
NB : pixel vient de «picture element»



74

74

Pixels et niveaux de gris



75

75

Échantillonnage spatial



76

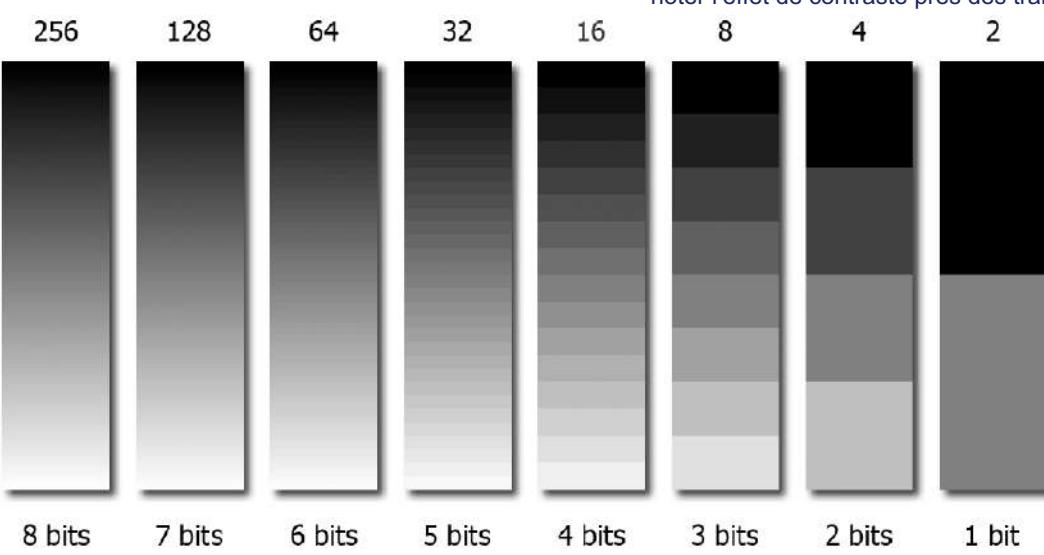
 256×256  128×128  64×64  32×32

 SORBONNE
UNIVERSITÉ

76

Quantification des luminances

noter l'effet de contraste près des transitions



 SORBONNE
UNIVERSITÉ

77

Quantification des luminances



Définition d'une image numérique



Définition d'une image numérique

Largeur
Hauteur
Indice de ligne
Indice de colonne
Le pixel $I_{i,j}$
 $I_{i,j} - N$
Une image numérique I

93	62	10	96	13	202
66	88	44	27	65	180
95	121	83	71	106	184
131	139	110	106	159	218
54	42	32	38	107	185
86	80	74	82	143	204
137	107	78	145	200	226
47	25	40	39	169	198
86	69	66	128	187	210
112	109	107	186	220	229
39	53	79	105	189	199
82	98	120	175	207	207
28	162	186	208	220	222
60	107	144	179	194	190
107	149	180	201	207	195
169	192	206	220	219	224
117	148	170	189	187	187
156	171	182	195	192	194

80

S^m SORBONNE UNIVERSITÉ

Image en couleurs

Le code Python suivant permet d'afficher les valeurs d'un pixel d'une image.

```
from PIL import Image
img = Image.open("chien.tif")
(r,v,b) = img.getpixel((18,12))
print("r : ", r, "v : ", v, "b : ", b)
```

Pour le pixel en position (18, 12), on obtient le triplet (96, 93, 62).

Canal Rouge

Canal Vert

Canal Bleu

82

```
<p><script>
function setColor()
{var r, g, b; r = document.getElementById("r01");
g = document.getElementById("g01");
b = document.getElementById("b01");
if (r.value.length == 0) {r.value = "0";}
if (g.value.length == 0) {g.value = "0";}
if (b.value.length == 0) {b.value = "0";}
if (isNaN(r.value)) {r.value = "0";}
if (isNaN(g.value)) {g.value = "0";}
if (isNaN(b.value)) {b.value = "0";}
if (parseInt(r.value) < 0) {r.value = "0";}
if (parseInt(g.value) < 0) {g.value = "0";}
if (parseInt(b.value) < 0) {b.value = "0";}
if (parseInt(r.value) > 255) {r.value = "255";}
if (parseInt(g.value) > 255) {g.value = "255";}
if (parseInt(b.value) > 255) {b.value = "255";}

document.getElementById("result01").style.backgroundColor =
= "rgb(" + r.value + ", " + g.value + ", " + b.value + ")";
setColor();
</script></p>
```

Image en couleurs

<https://interstices.info/tout-ce-que-les-algorithmes-de-traitement-dimages-font-pour-nous/>

83



Formats d'une image numérique

2 types d'images sont utilisés en informatique :

a) Les images (matricielles) ou bitmap :

- Image constituée d'un ensemble de points : les pixels.
- Chaque point porte des informations de position et de couleur.
- Format d'images bitmap : BMP, GIF, JPEG, TIFF, PCX.
- Les photos numériques et les images scannées sont de ce type.

b) Les images vectorielles :

- Composées de formes géométriques qui vont pouvoir être décrites d'un point de vue mathématique.
- Par exemple une droite sera définie par 2 points, un cercle par un centre et un rayon.
- Le processeur est chargé de "traduire" ces formes en informations interprétables par la carte graphique (images Word, Publisher, CorelDraw - format WMF, CGM, etc.)

NB :

Les avantages d'une image vectorielle : les fichiers qui la composent sont petits, les redimensionnements sont faciles sans perte de qualité.

Les inconvénients : une image vectorielle ne permet de représenter que des formes simples. Elle n'est pas donc utilisable pour la photographie notamment pour obtenir des photos réalistes.

84

Bitmap vs. Vectoriel

Agrandissement en bitmap



85



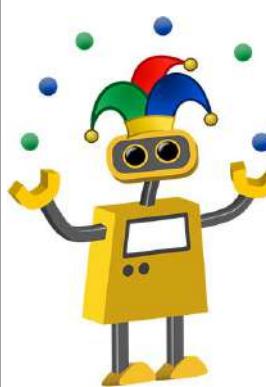
Agrandissement en vectoriel



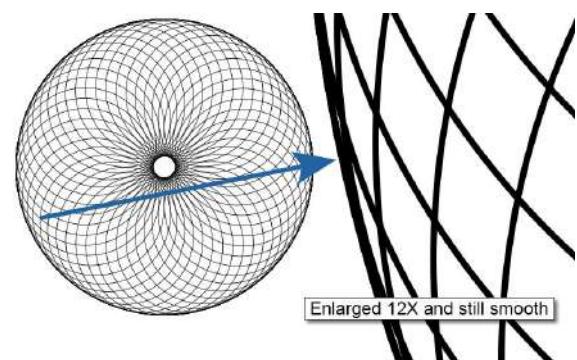
SORBONNE
UNIVERSITÉ

85

Bitmap vs. Vector images



Enlarged 12X
to show individual pixels



Enlarged 12X and still smooth

86

SORBONNE
UNIVERSITÉ

86

Vector vs. Bitmap images

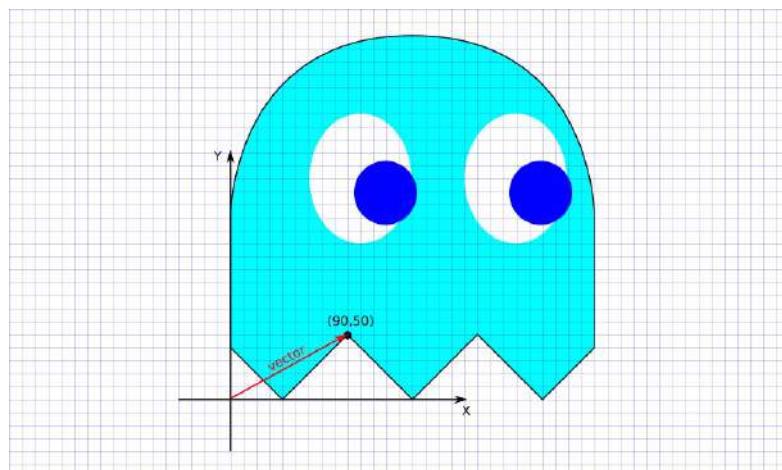


87



87

Vector vs. Bitmap images



This is how 2D vectors describe all points in 2D vector graphics.

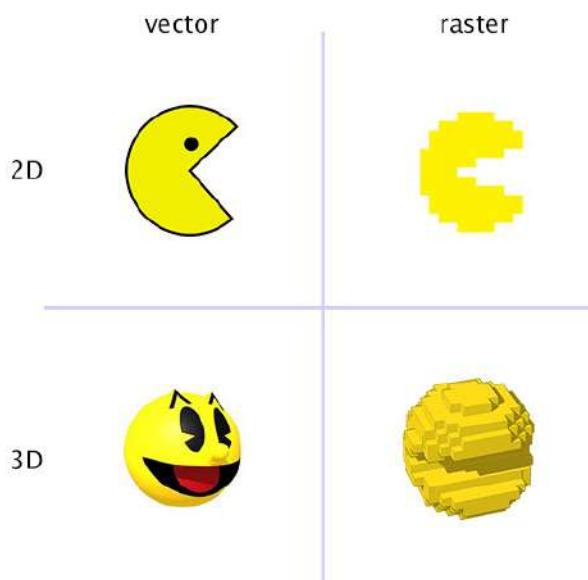
88



88

Vector vs. Bitmap images

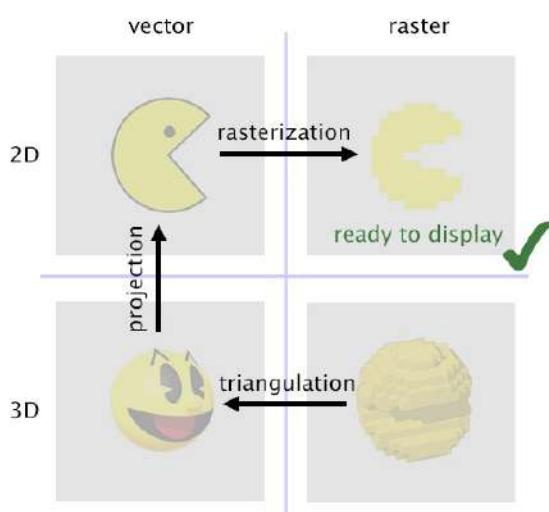
89



89

Vector vs. Bitmap images

90



La rastérisation (matricialisation) consiste à prendre une image décrite dans un **format graphique vectoriel** (formes) et à la convertir en une **image pixelisée** (une série de pixels, de points ou de lignes qui, lorsqu'ils sont affichés ensemble, créent l'image qui était représentée par les formes).

90

Représentation d'une image numérique

Représentation matricielle

Représentation lexicographique de l'image : matrice

$$I = [0 \dots M - 1] \times [0 \dots N - 1]$$

Largeur = nombre de colonnes N , hauteur = nombre de lignes M

Le pixel au croisement de la ligne i et de la colonne j désigné par $I(i, j)$

Représentation vectorielle

Lignes de l'image juxtaposées : vecteur

$$v = [0 \dots, M \times N - 1]$$

Le pixel (i, j) : composante $v[jN + i]$

92



92

Les formats BITMAP

Format	Points forts	Points faibles	Note
JPEG JPEG 2000 (Joint Photographic Experts Group)	Très forte compression	Compression destructrice	Spécialement conçu pour les photographies, il est cependant à utiliser avec délicatesse tant sa compression peut brouiller l'image. Le format JPEG2000, évolution du format original, peut être réglé pour compresser sans pertes.
GIF (Graphical Interchange Format)	Possibilité d'animation et de transparence, compression efficace	Limité à 256 couleurs	Très répandu sur le Web malgré ses faiblesses et un problème de droit sur son format de compression. À déconseiller pour les photos.
PNG (Portable Network Graphics)	Excellent compression sans perte. Possibilité de transparence. Standard donc pérenne.	Pas très efficace pour les larges photographies	Format destiné à remplacer le format GIF et ses limitations, mais ayant encore du mal à s'implanter dans les habitudes des développeurs. Peut remplacer les JPEG comme les GIF (sauf en ce qui concerne l'animation).
TIFF (Tagged Image File Format)	Compression sans perte efficace. Couche de transparence.	Lourdeur des fichiers non compressés. Format propriétaire.	Format de stockage très utilisé, à éviter pour le Web
BMP (Bitmap)	Format par défaut de Windows	Disponible uniquement sur la plateforme de Microsoft	Généralement non compressé et de ce fait des fichiers très « lourds »

93



93

Formats d'une image numérique

Les formats de fichiers des appareils photos numériques (APN)

- Les images sauvegardées sur la carte mémoire d'un APN sont **toujours de type Bitmap**.
- Le format le plus répandu est le format Jpeg : c'est un format de compression qui peut compresser les fichiers à plus de 90 %.
- Les APN sont accompagnés de logiciels qui permettent de faire un choix du taux de compression.
- Plus le taux de compression est élevé plus l'image est altérée.
- Inversement, un taux de compression moins grand donne une qualité supérieure d'imag, mais un fichier plus gros.

94



94

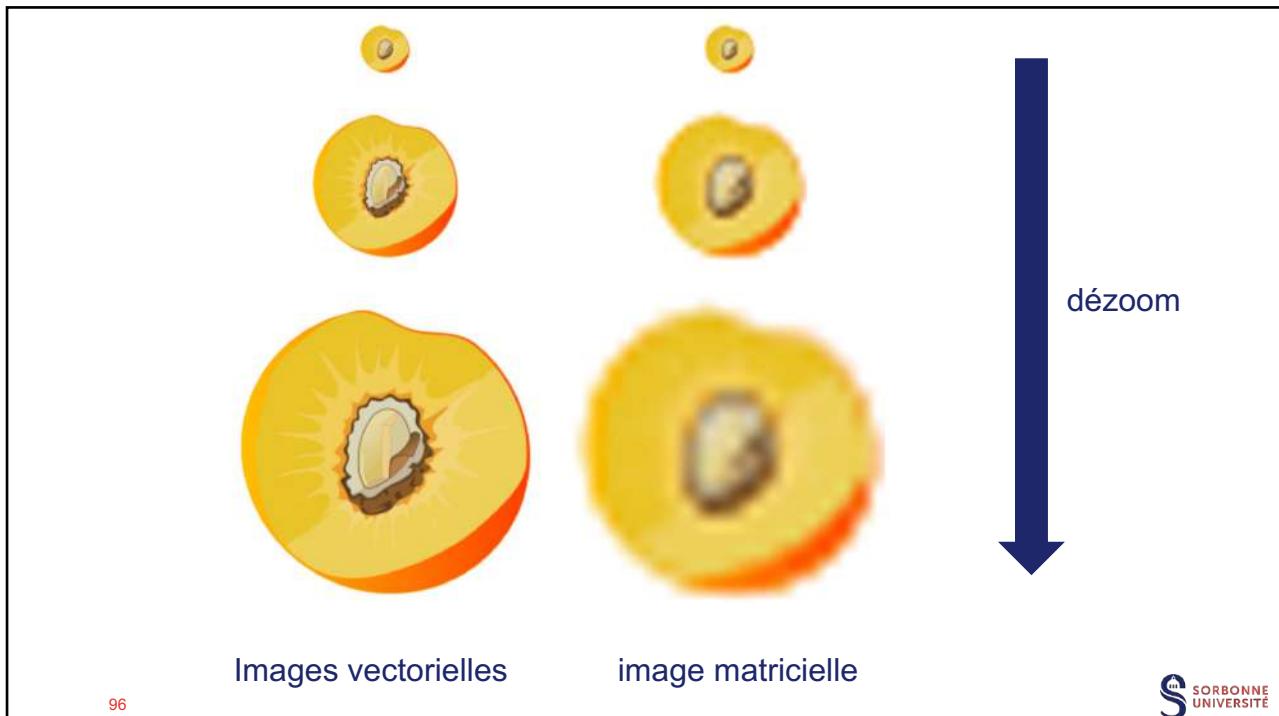
Les formats Vectoriels

Format	Points forts	Points faibles	Note
AI (Adobe Illustrator)	Reconnu par tous les logiciels graphiques.	Format propriétaire.	Format standard de Adobe Illustrator, l'un des plus utilisés du fait de la popularité du logiciel.
PS/EPS (Postscript / Encapsulated Postscript)	Très bien reconnu sur tous les systèmes.	N'a d'intérêt que dans le cadre d'une impression. Fichier très lourd.	Format hybride bitmap/vectoriel, réservé à l'impression. EPS est un fichier PS qui comporte quelques restrictions supplémentaires.
SVG (Scalable Vector Graphics)	Format XML donc extensible. Très compressible car format texte. Standard donc pérenne. Permet les animations et la transparence. Peut afficher des images bitmap.	Encore très peu reconnu, car peu d'outils disponibles et manque d'implémentation au sein de navigateurs (besoin d'un plugin).	Promis à un grand avenir malgré un démarrage lent, ce format est souvent cité comme capable de rivaliser avec les premières versions de Flash.
FLA / SWF (Flash)	Très polyvalent, peut utiliser des mp3, des JPEG, des vidéos... Très répandu sur le Web.	Format propriétaire et fermé.	C'est le standard de fait des animations vectorielles sur le Web.
PDF (Portable Document Format)	Affiche les documents	Taille prohibitive. Ne peut se lire qu'avec le logiciel Acrobat ou logiciel équivalent.	Version simplifiée de PostScript, il a été conçu pour afficher les documents de la même manière quel que soit le système.
PICT (Picture)	Format par défaut de Mac OS, donc encore utilisé.	Disponible uniquement sur la plateforme d'Apple	N'a plus grand intérêt face aux autres formats existants.

95



95



96



Résolution et définition d'une image

Les termes Résolution et Définition sont souvent confondus dans le langage de la photo numérique. Ainsi le menu de configuration d'un APN donnera plusieurs choix de réglage de la **Résolution** d'une image alors que l'on devrait plutôt parler de **Définition**.

Les mots Résolution et Définition sont liés à la notion de pixels.
Le pixel est l'élément de base d'une image ou d'un écran, c'est-à-dire un point.

La définition est le nombre de points (ou pixels) que comporte une image numérique en largeur et en hauteur. On l'exprime en donnant le nombre de pixels en hauteur et en largeur (exemple : 1600x1200). Ne pas confondre avec la résolution.

La résolution est le nombre de pixels par unité de longueur.
La résolution est exprimée le plus souvent en ppp (point par pouces) ou en dpi (dots per inch).
Rappel : 1 pouce = 2.54 cm.
Une image de 100 ppp (ou 100 dpi) contient 10 000 points par pouce carré ($100 \times 100 = 10 000$).

NB : La résolution définit la netteté d'une image et sa qualité d'affichage à l'écran. Plus la résolution est grande (c'est-à-dire plus il y a de pixels dans une longueur de 1 pouce), plus votre image est précise dans les détails. C'est un critère important lorsque l'on utilise un scanner.

97



97

Qualité de l'image

Lignage : phénomène d'alternance des lignes claires (ou sombres) de même direction et qui tranchent avec le reste de l'image

Contraste : qualité de la dynamique des intensités de l'image

Bruit : signal "parasite" dont la distribution dans l'image est aléatoire et la plupart du temps, inconnue

Déformations géométriques : défauts dus à la différence d'axe entre le capteur d'acquisition et le centre de la scène observée

98



98

Qualité des images: propriétés

Moyenne : moyenne des niveaux de gris de l'image (on l'appelle aussi **brillance** ou **luminance**)

$$Moy = \frac{1}{NM} \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} I(x, y)$$

Contraste, plusieurs définitions possibles :

- Ecart-type des variations de niveaux de gris :

$$C = \sqrt{\frac{1}{NM} \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} (I(x, y) - Moy)^2}$$

- Variations entre valeurs de niveaux de gris min et max :

$$C = \frac{\max\{I(x, y)\} - \min\{I(x, y)\}}{\max\{I(x, y)\} + \min\{I(x, y)\}}$$

99



99

Contenu de l'image

Texture : répartition statistique ou géométrique des intensités dans l'image

Contour : limite entre deux (ou un groupe de) pixels dont la différence de niveau de gris (couleur) est significative

Région : groupe de pixels présentant des caractéristiques similaires (intensité, mouvement, etc.)

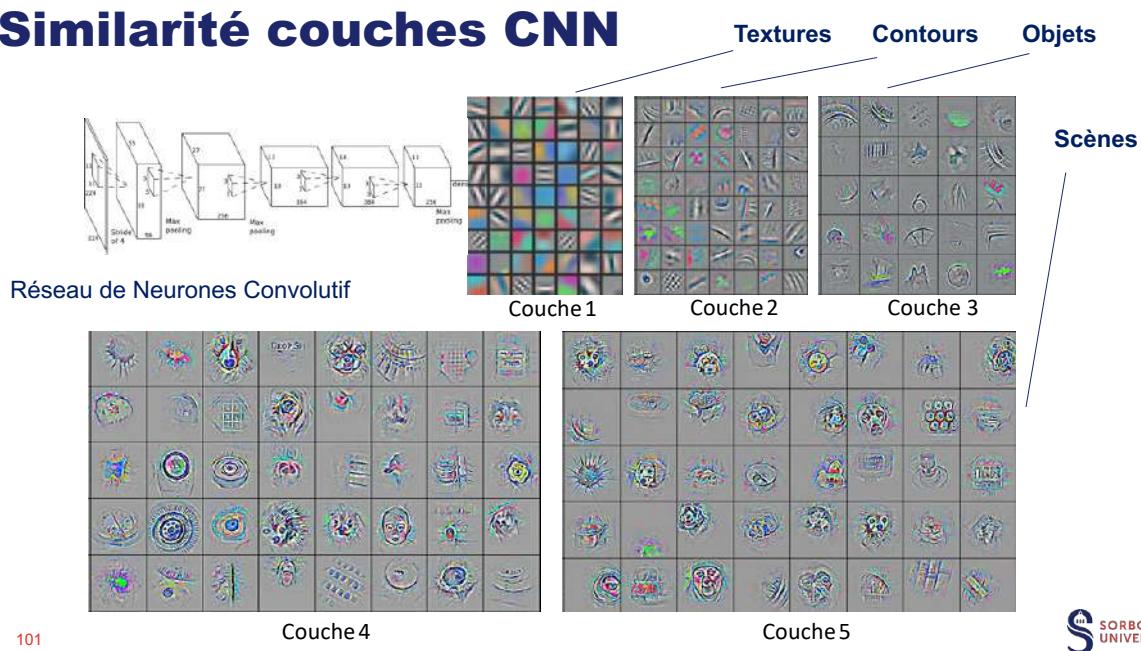
Objet : région (groupe de régions) entièrement délimitée par un contour, possédant une indépendance dans l'image

100



100

Similarité couches CNN



101



101

Voisinage de pixels

Voisinage = notion fondamentale en traitement d'images

- Algorithmes de suivi de contours, croissance de région (deux ou plusieurs pixels voisins peuvent être agrégés si leurs caractéristiques sont « proches »)

On parle de **connexité** d'un pixel :

Connexité d'ordre 4 (4-Connectivity) : on considère les 4 voisins directs N, S, O et E du pixel

Connexité d'ordre 8 (8-Connectivity) : on considère les 8 voisins directs N, NE, NO, S, SE, SO, O et E du pixel

La connexité peut s'étendre aux voisins indirects (i.e. pas uniquement ceux de la première couronne)

103



103

Voisinage V d'un pixel P

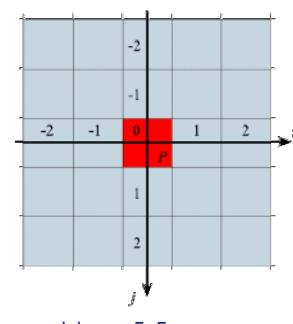
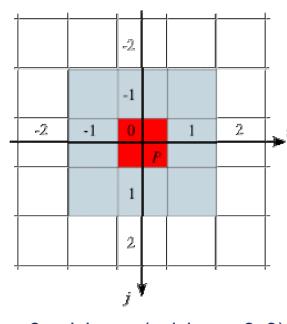
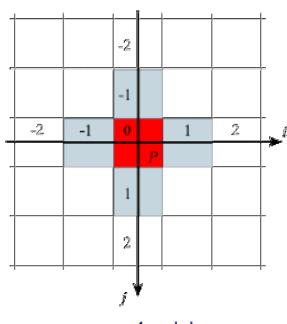
Hypothèses :

Le voisinage V est centré en P et les pixels sont disposés selon une maille carrée

Définition : $V(P)$ est l'ensemble des pixels Q situés à moins d'une certaine distance de P

La forme du voisinage (et le nombre de voisins) de P dépendent de la distance considérée.

Voisinages les plus usités en traitement d'images :



104



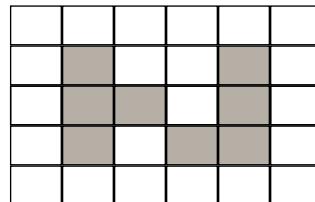
104

Connexité 4 vs. 8

Un pixel possède plusieurs voisins (4 ou 8)



Connexité 4 ou 8 :



Région grise :

UN seul objet en connexité 8
DEUX objets en connexité 4

105



105

La vision en couleurs

La vision des couleurs est la capacité d'un organisme ou d'une machine de distinguer des objets qui ne diffèrent que par le spectre de la lumière qu'ils réfléchissent.

La notion de couleur dépend de la vision humaine, qui, sauf cas de daltonisme, connaît le rayonnement électromagnétique entre **380 et 780 nanomètres** par un processus cognitif basé sur **trois types de récepteurs** dans la rétine.

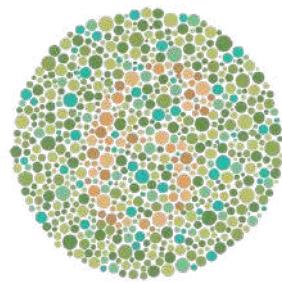
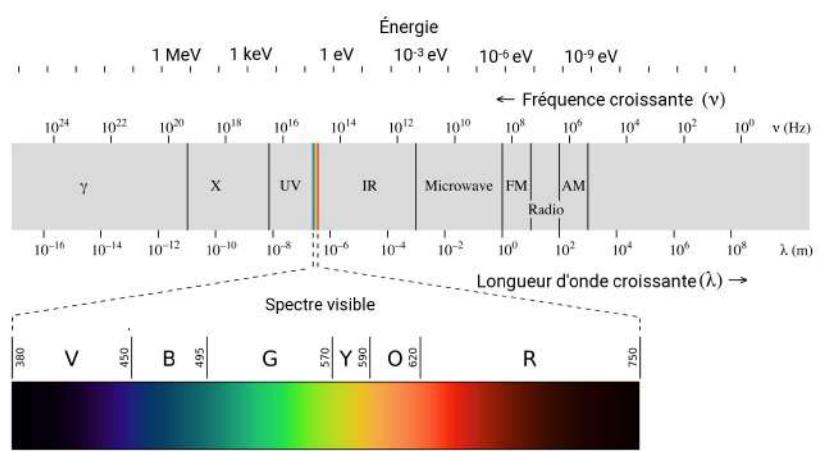
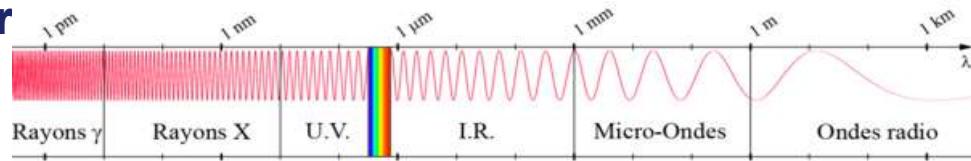


Planche du test d'Ishihara.
Seule la vision humaine en couleurs permet de lire un « 6 ».

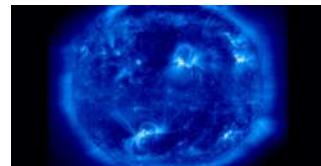


106

Longueur d'onde



rayons X



rayons UV



infra-rouges



micro-ondes



radio (IRM)



rayon gamma

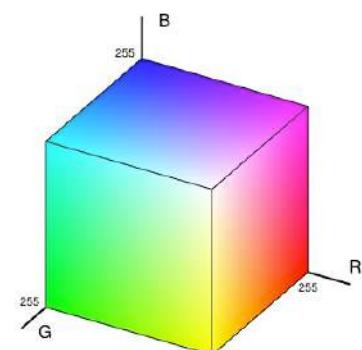
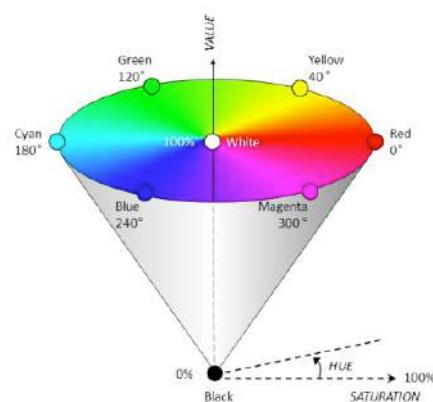
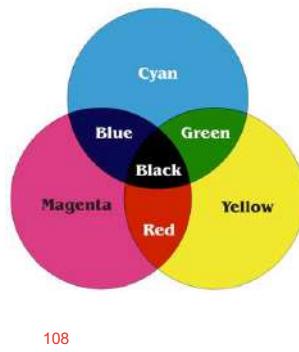
107

107

Espaces de couleurs

Une image peut être constituée de

- 1 bande à niveaux de gris \Rightarrow valeurs entre 0 et 255.
- 2 bandes – luminance, chrominance
- 3 bandes : les images couleur courantes : RGB (Red, Green, Blue), HSI (Hue, Saturation, Intensity), L*a*b, CMY (Cyan, Magenta, Yellow)



108

108

Courbe tonale (Look Up Table - LUT)

Retoucher une image revient à modifier les valeurs de certains pixels.

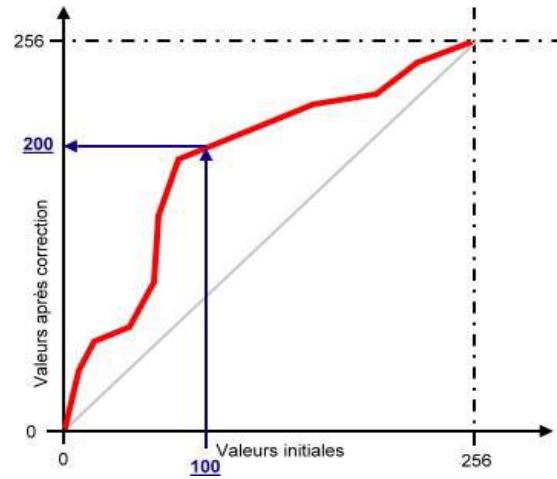
On peut le faire localement ou globalement.

Dans ce dernier cas, on utilise un outil appelé « courbe tonale », qui ressemble au dessin ci-contre.

Sur l'abscisse, on lit les valeurs originales des pixels et sur l'ordonnée les valeurs après modifications. Sur le graphique ci-contre, tous les pixels de valeurs 100 prendront la valeur 200. Ils vont donc s'éclaircir. La diagonale grise est la courbe où il n'y a aucune modification.

En fait, il y a trois courbes tonales : une pour le rouge, une pour le vert et une pour le bleu. On les modifie souvent simultanément de la même façon, mais on peut aussi les modifier séparément et différemment.

109

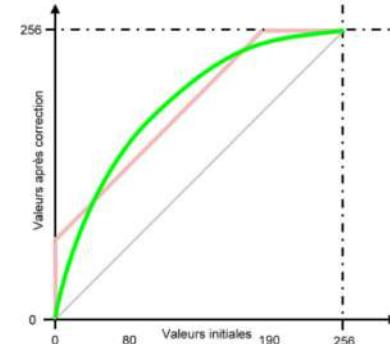
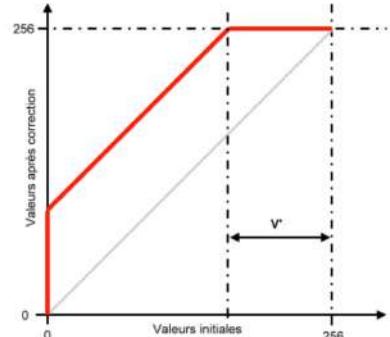


109

Changement de luminosité

Eviter une perte d'information
Réversibilité

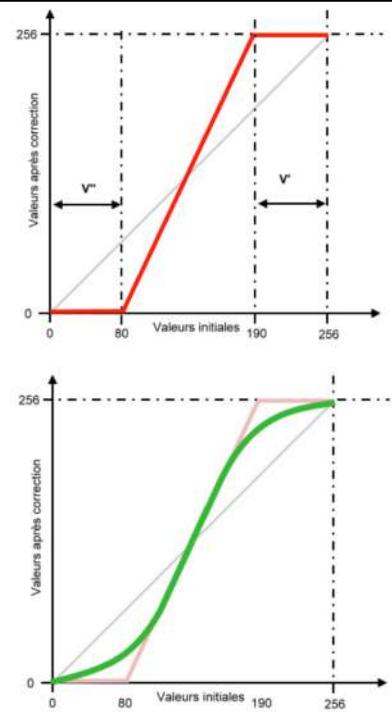
$$\mathbb{I}^j(x, y) = \mathbb{I}(x, y) + \beta$$



110

Augmentation de contraste

Eviter une perte d'information
Réversibilité



111

Négatif



112

112

Rouge



113



113

Niveaux de gris



114

$$\text{gris} = \text{int}(\text{round}(0.299 \cdot \text{rouge} + 0.587 \cdot \text{vert} + 0.114 \cdot \text{bleu}))$$


114

Seuillage

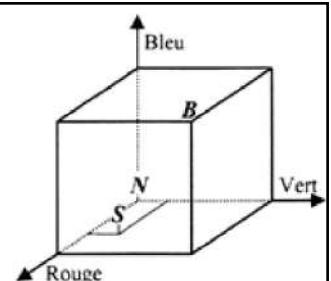
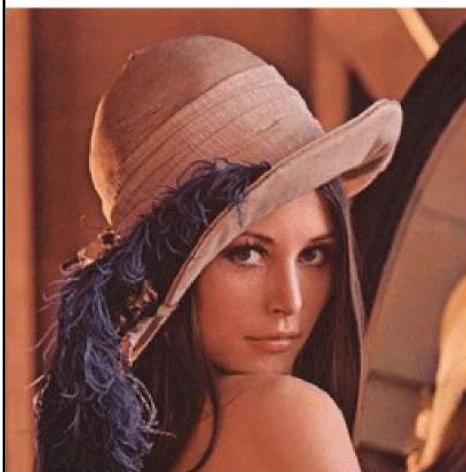


115



115

Sepia



En photographie, le sépia est une qualité de tirage qui ressemble au noir et blanc, mais avec des variations de brun, et non de gris.

Sépia dans le système RVB est S(94, 38, 18).



116

Sepia



117



117

Pixellisation



118



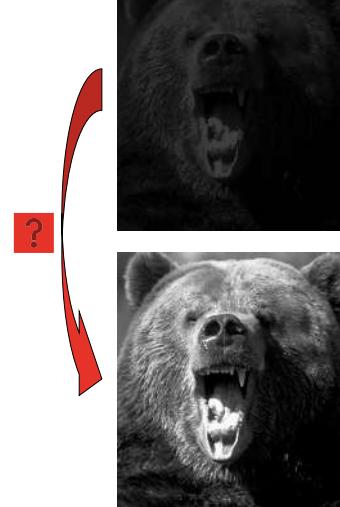
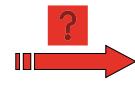
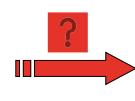
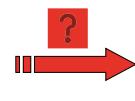
118

Modèles mathématiques fondamentaux



119

Possible, pas possible ?



120



120

Modèles et outils fondamentaux

Introduction aux outils d'analyse d'images numériques les plus courants.

Raisons didactiques - présentation organisée selon les principaux *modèles mathématiques* qui ont été employés pour traiter les images. Ces différents modèles ne sont cependant *ni exclusifs, ni cloisonnés*.

Aux différents modèles présentés correspondent un certain nombre d'*outils fondamentaux*, qui se sont révélés au cours du temps plus ou moins incontournables, que ce soit d'un point de vue pratique ou théorique.

Citons : la *convolution*, la *transformée de Fourier*, l'*histogramme*, la *corrélation*, la *transformée en tout-ou-rien*, les *ondelettes*...

Introduction aux outils les plus basiques ou les plus courants :

121



121

Plan de la première partie

1. INTRODUCTION

Intérêt du traitement d'image
Applications du traitement d'images
Vocabulaire de base de l'image et du traitement de l'image

2. MODÈLES FONDAMENTAUX

- a. Modèle statistique (histogramme, entropie ...)
- b. Modèle linéaire (convolution)
- c. Modèle fréquentiel (Fourier ...)
- d. Modèle différentiel (gradient, EDP ...)
- e. Modèle ensembliste (morphologie ...)
- f. Modèle discret (maillage, connexité, distance ...)

122



122

a. Modèles statistiques



Traitement d'images

123

Modèle statistique

Dans ce modèle, on s'intéresse aux propriétés statistiques des images : la **distribution** des valeurs prises par les pixels, la **corrélation** entre des pixels spatialement proches, la **fréquence d'occurrence** de certaines structures spatiales...

Les mesures statistiques fournissent des grandeurs et fonctions empiriques sur lesquelles peuvent s'appuyer des modèles probabilistes utilisés par les algorithmes d'analyse d'images :

- **Champs de Markov** : chaque pixel correspondant à une variable aléatoire dont la valeur ne dépend que de celle de ses voisins
- **Matrices de cooccurrence** : texture
- ...

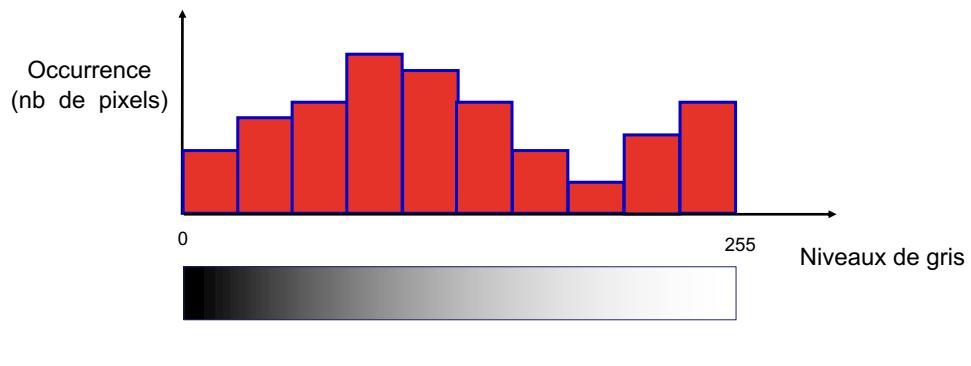


124

124

Modèle statistique : l'histogramme

L'histogramme représente la **répartition** des pixels en fonction de leur **niveau de gris**. Il fournit diverses informations comme les statistiques d'ordre, l'entropie, et peut permettre d'isoler des objets.

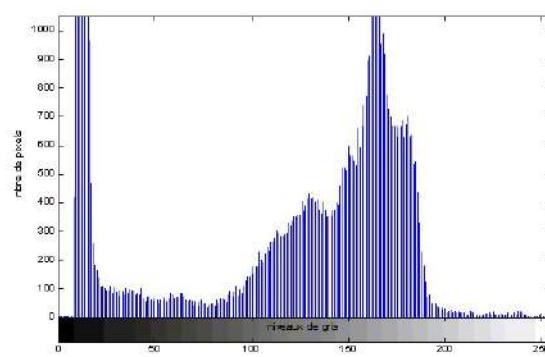


125

Histogramme

L'histogramme d'une image est une fonction discrète. Elle représente le nombre de pixels en fonction du niveau de gris et suit la forme du graphe de densité de probabilité des niveaux de gris de l'image avant leur numérisation.

Lorsque cette fonction est normalisée entre 0 et 1 pour tous les niveaux de gris, on peut la voir comme une **densité de probabilité** qui fournit la probabilité de trouver un certain niveau de gris de l'image. Ainsi le niveau de gris d'un pixel devient une variable aléatoire dont la valeur dépend du résultat d'une **expérience aléatoire sous-jacente**. D'où un traitement statistique des images.



126

Histogramme

Définition : Fonction décrivant la répartition des niveaux de gris de l'image

Fournit des informations importantes sur l'image, telles que :

- la distribution statistique des niveaux de gris,
- les bornes de répartition des niveaux de gris.

Mais aucune information spatiale !

A chaque image I de taille $N \times M$, on associe une distribution H des valeurs contenues dans cette image par :

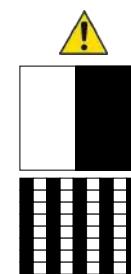
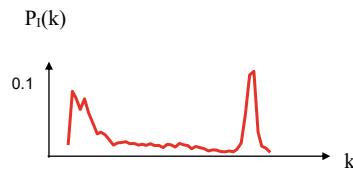
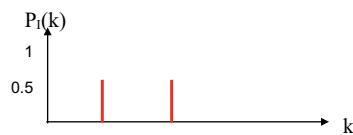
$$H(k) = \underbrace{\text{card}\{0 \leq x \leq N-1, 0 \leq y \leq M-1 : I(x, y) = k\}}_{n_k}$$

127



127

Modèle statistique : l'histogramme



Deux images différentes peuvent avoir le même histogramme

128

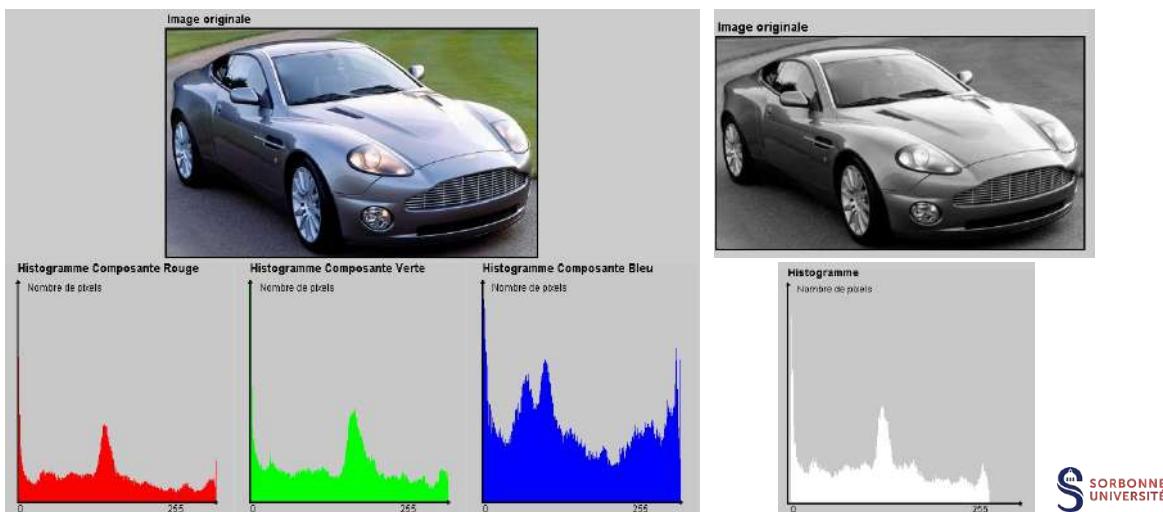


128

Modèle statistique : l'histogramme

Histogramme couleur :

pour une image couleur, on traite chaque canal indépendamment.

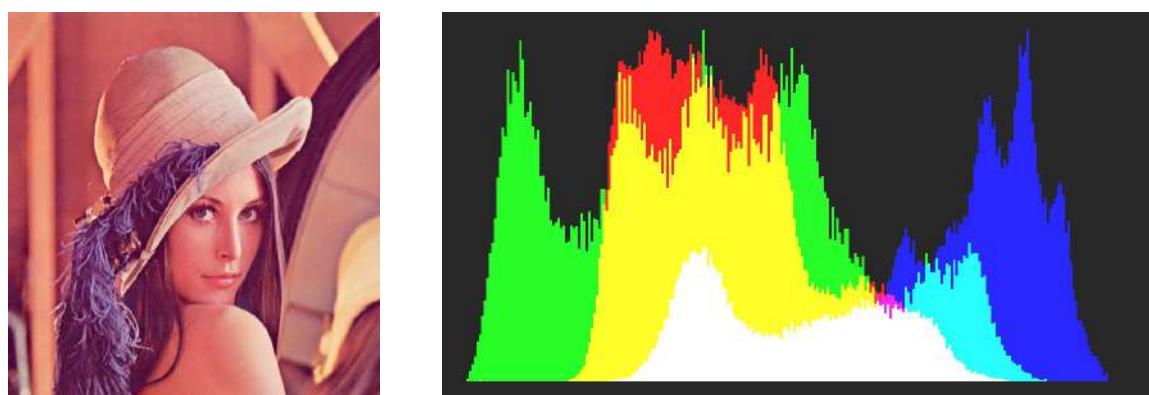


129

Modèle statistique : l'histogramme

Histogramme couleur :

pour une image couleur, on traite chaque canal indépendamment.



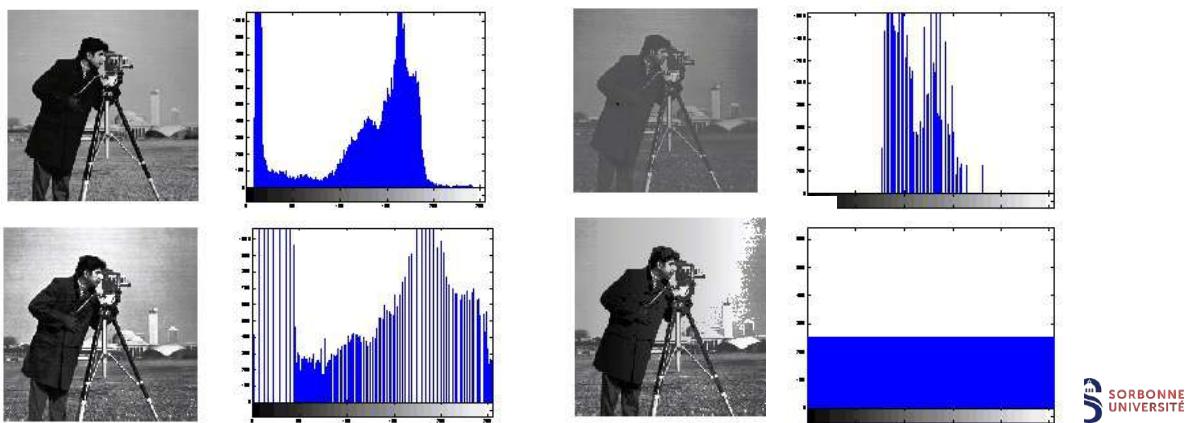
130

130

Sensibilité de l'image aux modifications d'histogrammes

Estournet a fait réaliser plusieurs copies d'une même photo sur des supports variés : papiers photographiques de marques, de gradations et de duretés différentes.

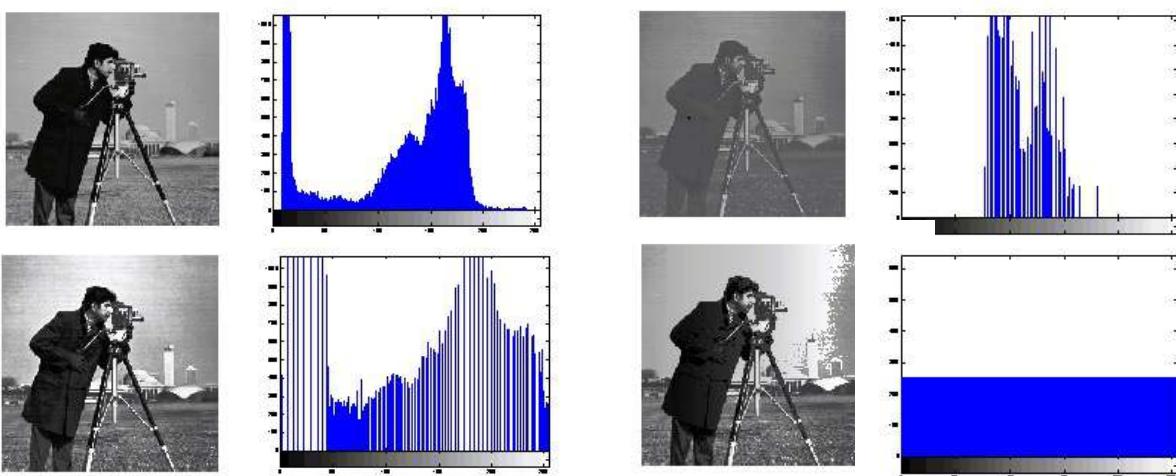
L'analyse de ces documents montre de **grandes différences d'histogrammes**, différences bien plus remarquables que celles qui affectent les images elles-mêmes.



131

Sensibilité de l'image aux modifications d'histogrammes

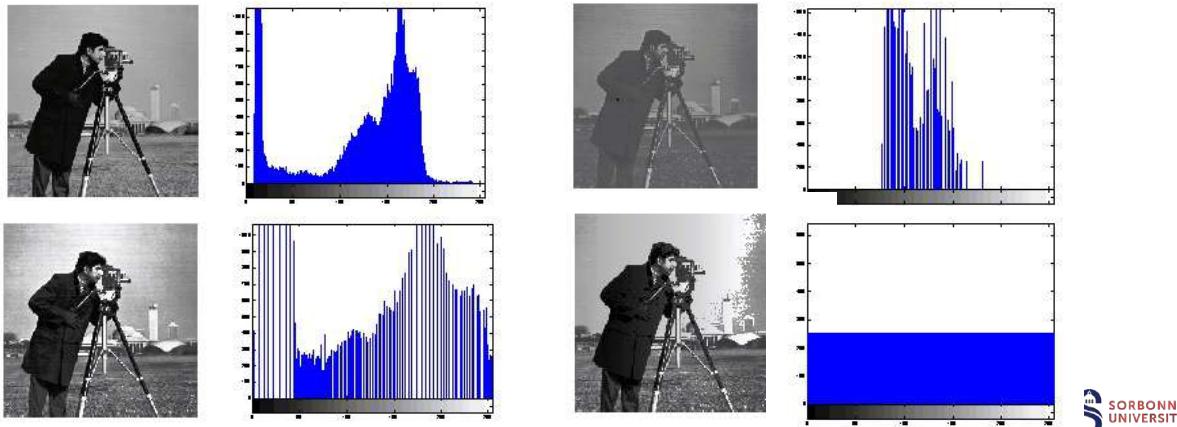
On en déduit que l'histogramme de l'image peut être modifié artificiellement sans que l'aspect extérieur de l'image n'en soit beaucoup affecté, et surtout sans que la signification de l'image ne soit altérée.



132

Sensibilité de l'image aux modifications d'histogrammes

NB : il faut s'imposer une contrainte pour que l'aspect de l'image demeure relativement inchangé : **Les relations d'ordre entre niveaux de gris doivent être conservées** : i.e. si n_1 et n_2 sont 2 niveaux de gris vérifiant $n_1 < n_2$, alors on les transformera en m_1 et m_2 tels que $m_1 < m_2$.



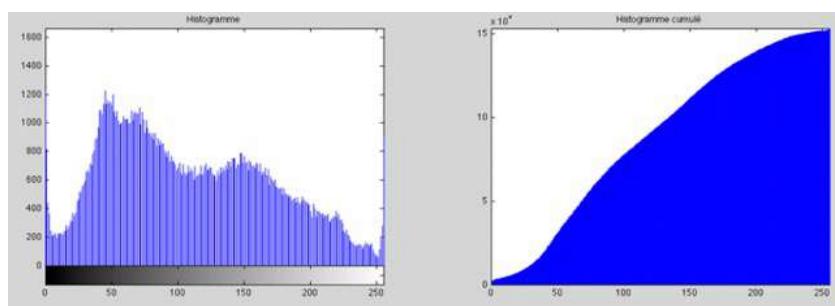
133

Histogramme cumulé

H_c représente la probabilité $P(i)$ du niveau i , soit la probabilité d'avoir un niveau inférieur ou égal à i :

$$H_c(k) = \sum_{i \leq k} H_n(i) = \frac{\sum_{i \leq k} H(i)}{N \times M}$$

où $H_n(\cdot)$ est l'histogramme normalisé et $H(\cdot)$ est l'histogramme initial
Fonction croissante qui tend vers 1.



¹³⁴ Le niveau de gris médian est x_m tel que $H_c(x_m)=0.5$.



134

Entropie

Suivant les idées de Shannon, l'**entropie** est une mesure statistique du contenu d'information d'un message. Elle est donc adaptée à caractériser les images, chacune d'entre elles étant prise comme un message spécifique.

En suivant cette idée, Hartley (1925) attribue à un symbole i émis par une source stationnaire avec une probabilité p_i , la quantité d'information :

$$I_i = -\log(p_i)$$

Shannon définit alors l'information moyenne par symbole ou entropie par :

$$S = - \sum_{i=1}^n p_i \log(p_i)$$

135



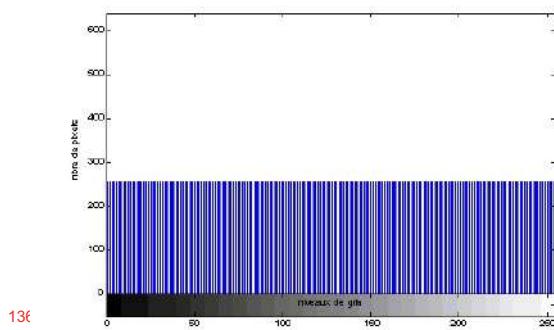
135

Entropie d'ordre 0

Considérant une image comme une source stationnaire de pixels, l'entropie se mesure donc à partir du seul **histogramme de l'amplitude**.

Utilisant les propriétés de la fonction $\log(x)$, on montre que l'entropie est maximale pour une répartition uniforme de probabilité $S_{max} = \log(n)$.

En terme d'histogramme, cela se traduit par un histogramme rigoureusement plat :



136

C'est cette notion d'entropie qui va être le fondement de la plupart des algorithmes présentés.

En effet les méthodes vont majoritairement viser à maximiser l'entropie d'ordre 0.



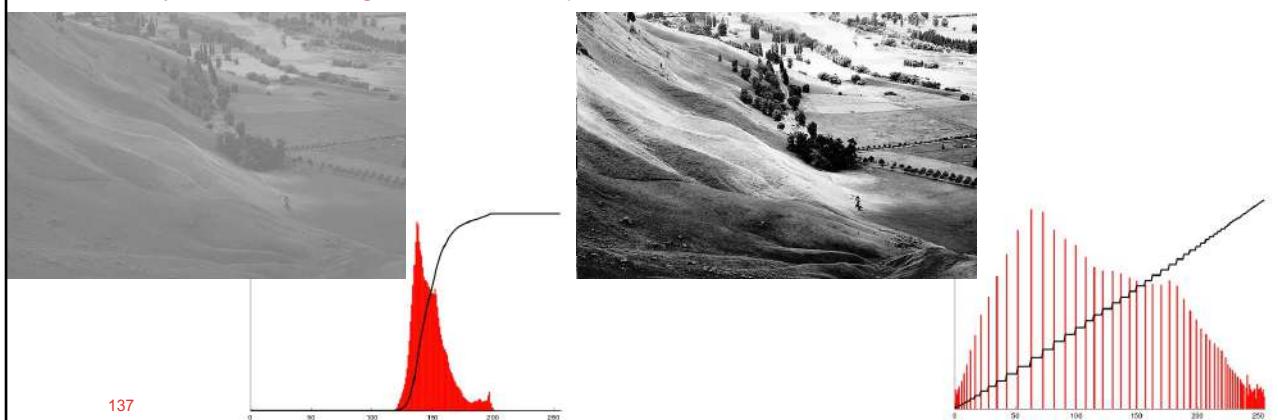
136

Pourquoi est-il nécessaire de modifier l'histogramme ?

Supposons que l'on ne puisse discerner que très peu de détails dans une image.

La raison la plus probable est que les pixels qui représentent **différents objets** ou différentes parties d'un objet ont un **niveau de gris très proche**.

Ceci est illustré par la figure suivante : l'histogramme de l'image « mauvaise » est très resserré alors que celui de l'image « bonne » est plus étalé.



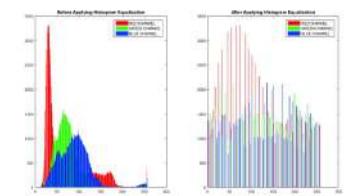
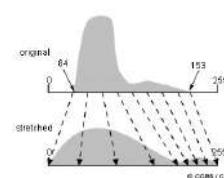
137

Modèle statistique : l'histogramme

Traitements d'image classiques à partir de l'histogramme :

i) Normalisation d'histogramme :

Exploitation de toute la dynamique de codage



ii) Egalisation d'histogramme :

Équilibrer la dynamique du codage et augmenter le contraste

iii) Segmentation d'histogramme, dont le célèbre seuillage :

Simplifier l'image en regroupant les pixels selon leur valeur

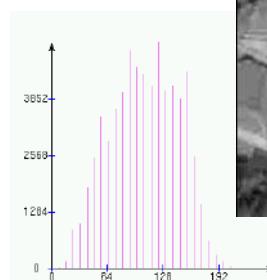
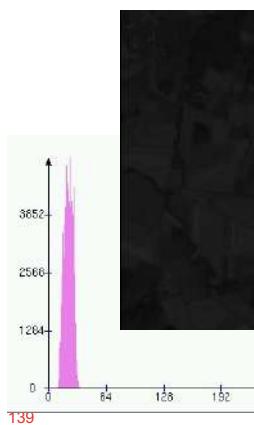


138

138

i) Normalisation d'histogramme

La **normalisation d'histogramme**, ou **expansion de dynamique**, est une **transformation affine** du niveau de gris des pixels de telle sorte que l'image utilise **toute la dynamique** de représentation.



i) Normalisation d'histogramme

Fonction H_n donnant la probabilité qu'un pixel ait pour le niveau de gris k

$$H_n(k) = \frac{H(k)}{N \times M}$$

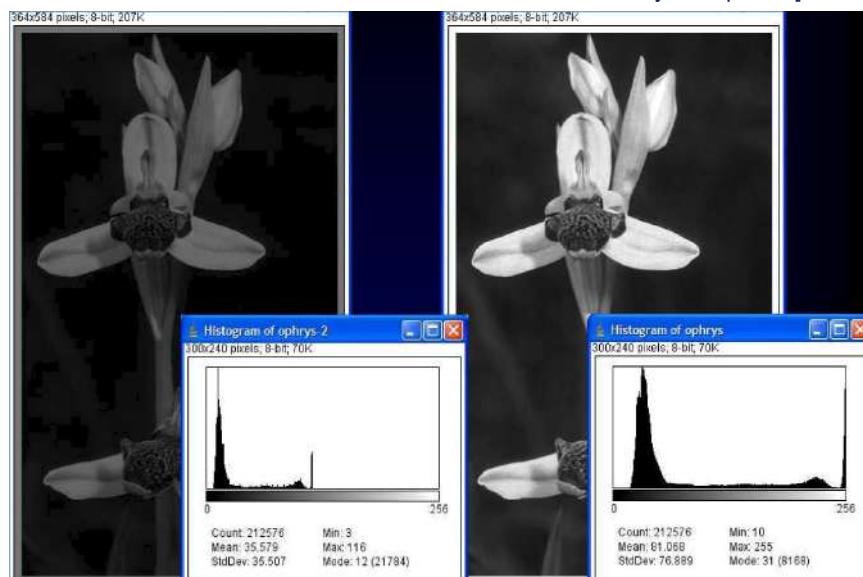
où N et M sont respectivement le nombre de colonnes et de lignes de l'image.

Approximation de la fonction de densité d'une variable aléatoire (pixel)

Les valeurs de H_n sont normalisées.

i) Normalisation d'histogramme

Dynamique = [valeur mini, valeur maxi]



i) Normalisation d'histogramme

$$I^j(x, y) = \frac{I(x, y) - I_{min}}{I_{max} - I_{min}} \cdot 255$$

Implémentation - version 1

```
for ( x=0; x < width ; ++ x )
    for ( y=0; y < height ; ++ y )
        I ( x,y ) = 255 * ( I ( x,y ) - Imin ) / ( Imax - Imin )
```

→ cela génère beaucoup d'opérations, à effectuer pour chaque pixel.

i) Normalisation d'histogramme

$$I^j(x, y) = \frac{I(x, y) - I_{min}}{I_{max} - I_{min}} \cdot 255$$

Implémentation - version 2 : la LookUp Table (LUT)

```
for ( i=0; i < 255; ++ i )
    LUT [ i ]= c l a m p ( 255 * ( i - Imin ) / ( Imax - Imin ) , 0 , 255 )

for ( x=0; x < width ; ++ x )
    for ( y=0; y < height ; ++ y )
        I ( x , y ) = LUT [ I ( x , y ) ]
```

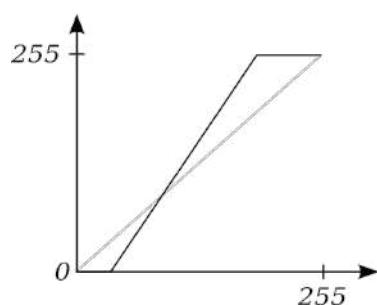
↔ on perd du temps à l'initialisation
 ↔ on le regagne largement pendant le traitement

143



143

LookUp Table (LUT)



LUT :

- une table de correspondance.
- une fonction injective.
 ↔ à chaque valeur de départ correspond une valeur d'arrivée

144



144

LookUp Table (LUT)

Avantages :

- plus rapide à appliquer à toute l'image
- on peut cumuler des LUT
 - en travaillant sur des flottant
↔ interpolation linéaire sur la lecture de la LUT
 - en autorisant les dépassemement
↔ le clamp est fait tout à la fin
- générique, quelque soit la transformation

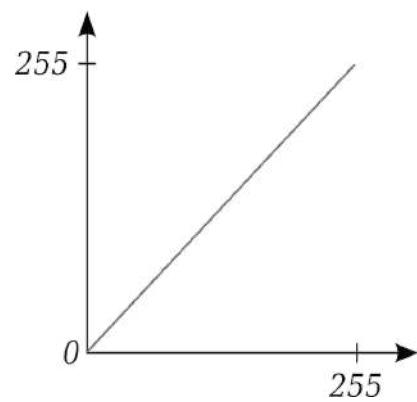
145



145

Combinaison de LUT

Image initiale :



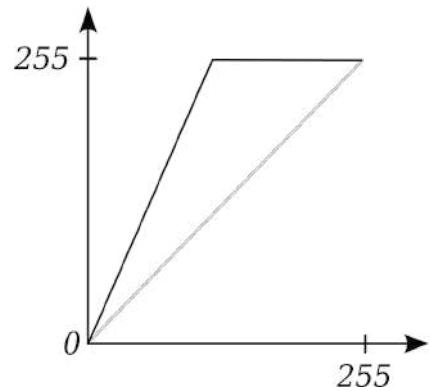
146



146

Combinaison de LUT

Augmentation de la dynamique :

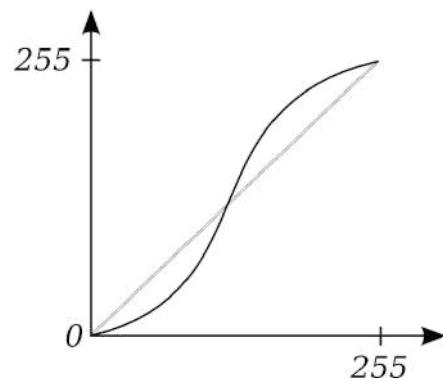


147

147

Combinaison de LUT

Augmentation du contraste :

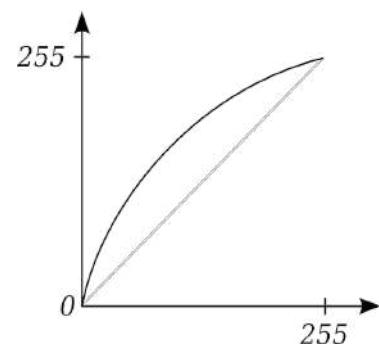


148

148

Combinaison de LUT

Correction gamma :

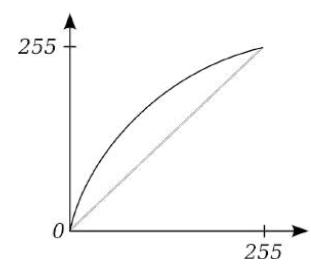
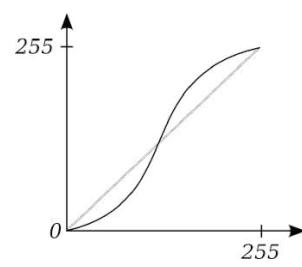
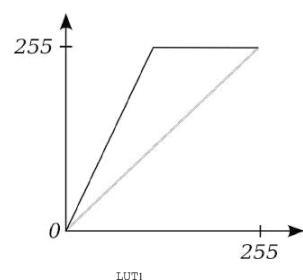


149



149

Combinaison de LUT

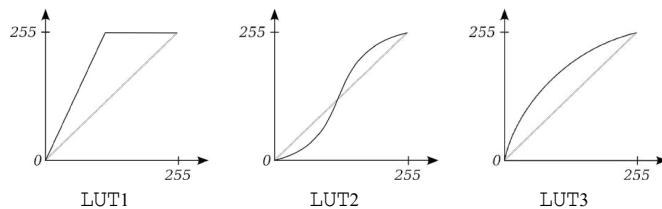


150

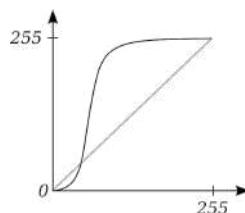


150

Combinaison de LUT



$$\text{LUT4} = \text{LUT3} \circ \text{LUT2} \circ \text{LUT1}$$



151



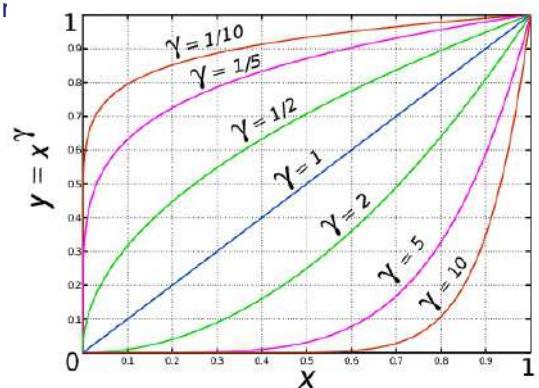
151

Correction gamma

Une caméra capture une image, qui s'affiche ensuite sur votre écran. L'image prise par l'appareil photo n'est pas toujours la même que celle affichée sur votre écran.

Chaque moniteur a une courbe de réponse intensité / tension qui est à peu près une fonction de puissance 2,5. Cela signifie que lorsque vous fournissez une tension de x , l'intensité est de x à la puissance 2,5. Par conséquent, le moniteur afficherait l'image plus sombre qu'elle ne l'est réellement.

Pour tenir compte de cette erreur, nous devons introduire une correction, appelée **correction gamma**, dans le signal d'entrée du moniteur.



156



156

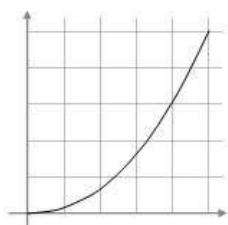
Correction gamma

 γ - Monitor

Correction couleur

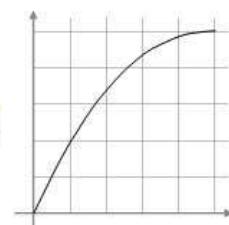


Résultat final



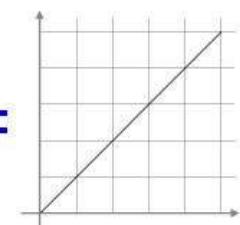
Gamma characteristics of monitors

X



Color information adjusted to match gamma characteristics

=

Color handling approaching the " $y = x$ " ideals

157



157

Correction gamma

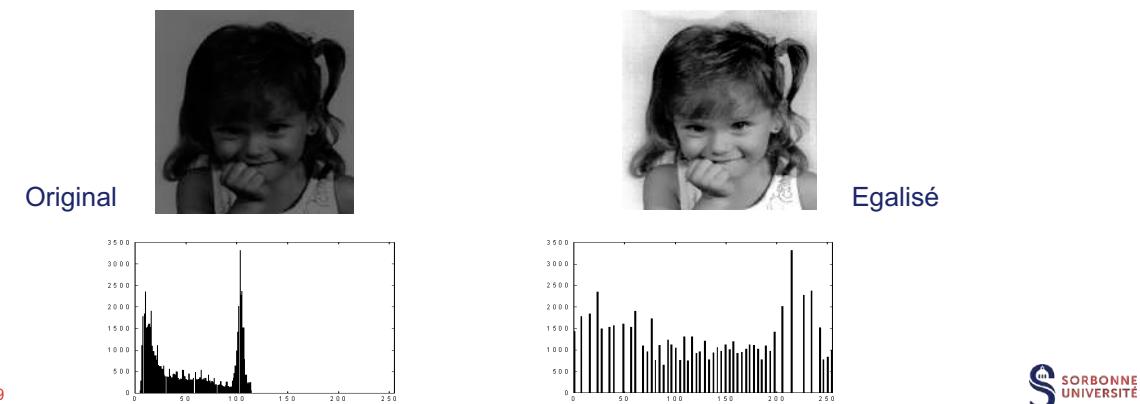
 $\gamma=2$  $\gamma=1/3$  $\gamma=1$ (original) $\gamma=1/4$  $\gamma=1/4$

158

ii) Egalisation d'histogramme

L'**égalisation d'histogramme** consiste à harmoniser la répartition des niveaux de luminosité de l'image de telle manière à tendre vers un même nombre de pixel pour chacun des niveaux de l'histogramme.

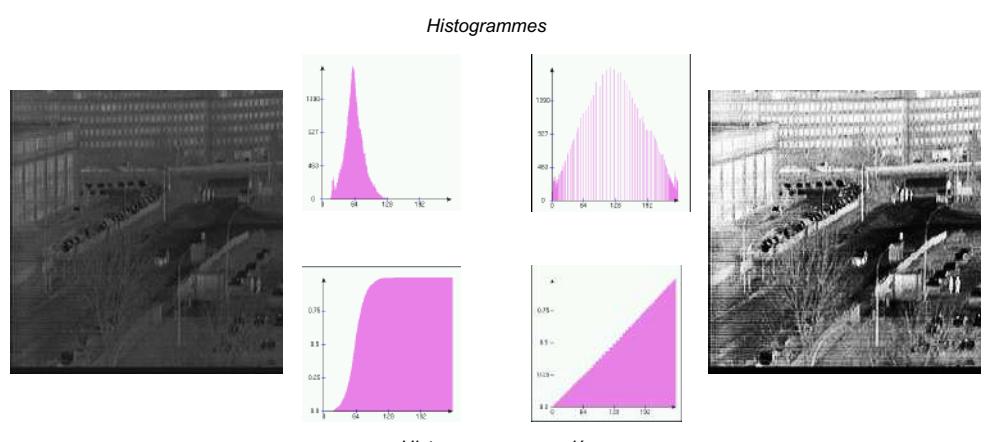
Cette opération vise à augmenter les nuances dans l'image



159

ii) Egalisation d'histogramme

La technique classique consiste à rendre « le plus linéaire possible » l'histogramme cumulé de l'image



160

160



ii) Egalisation d'histogramme

La méthode consiste à appliquer une **transformation T** indépendamment, sur chaque pixel de l'image. Cette transformation est construite à partir de **l'histogramme cumulé**.

Pour une image $\{x\}$ en niveaux de gris codée sur L niveaux, on définit n_k le nombre d'occurrences du niveau x_k . La probabilité d'occurrence d'un pixel de niveau x_k dans l'image est:

$$p_x(x_k) = p(x = x_k) = \frac{n_k}{n}, \quad 0 < k < L$$

avec n le nombre total de pixels de l'image, et p_x l'histogramme normalisé sur $[0,1]$.

La transformation T qui à chaque pixel de valeur x_k de l'image d'origine associe une nouvelle valeur s_k , $s_k = T(x_k)$ est alors définie par

$$T(x_k) = (L - 1) \sum_{j=0}^k p_x(x_j)$$

où $\sum_{j=0}^k p_x(x_j)$ est l'histogramme cumulé



161

ii) Egalisation d'histogramme

Il est possible de donner une justification théorique à l'égalisation d'histogramme si on se place dans le cas continu.

Pour une image $\{x\}$ en niveaux de gris codée sur L niveaux, les intensités des pixels de l'image sont vues comme des variables aléatoires dans $[0, L-1]$.

On considère les densités de probabilités $p_x(x)$ et $p_s(s)$ de x et respectivement s , ainsi que la transformation T telle $s=T(x)$ avec T **continue, différentiable et strictement croissante**.

On a alors:

$$p_s(s) = p_x(x) \left| \frac{dx}{ds} \right|$$

162



162

ii) Egalisation d'histogramme

En choisissant T comme la fonction de répartition de la variable aléatoire x :

$$Y(x) = (L - 1) \int_0^x p_x(w) dw$$

alors on peut montrer que la densité de probabilité de s est uniforme et égale à :

$$p_s(s) = \frac{1}{L - 1}$$

L'application de la transformation T permet donc de transformer la densité de probabilité d'origine en une **loi uniforme**, ce qui est le but recherché dans l'égalisation d'histogramme, puisque tous les niveaux seraient utilisés, et avec une égale probabilité.

Toutefois, en pratique, c'est-à-dire avec des histogrammes discrets, il est rare d'obtenir des histogrammes plats, mais la méthode permet de **mieux répartir les intensités sur l'ensemble de la plage de valeurs**.

163



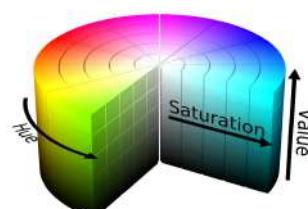
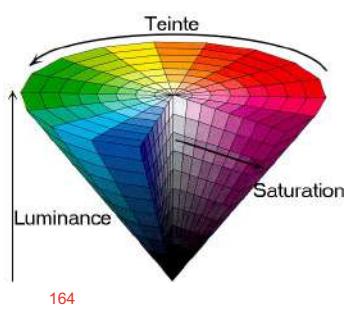
163

ii) Egalisation d'histogramme : images couleurs

Pour les images couleurs, il faut étendre la méthode d'égalisation en niveau de gris aux 3 composantes de couleur (i.e. *RGB*).

Il est possible de faire une égalisation indépendamment sur chaque composante, toutefois ceci dégrade les couleurs, et n'est donc pas utilisé en pratique.

La méthode utilisée est de réaliser l'égalisation uniquement sur les intensités, sans toucher aux couleurs. Ceci est fait en général dans **l'espace colorimétrique *HSV***, où l'égalisation est réalisée **sur la composante *V* uniquement (luminosité / luminance)**.

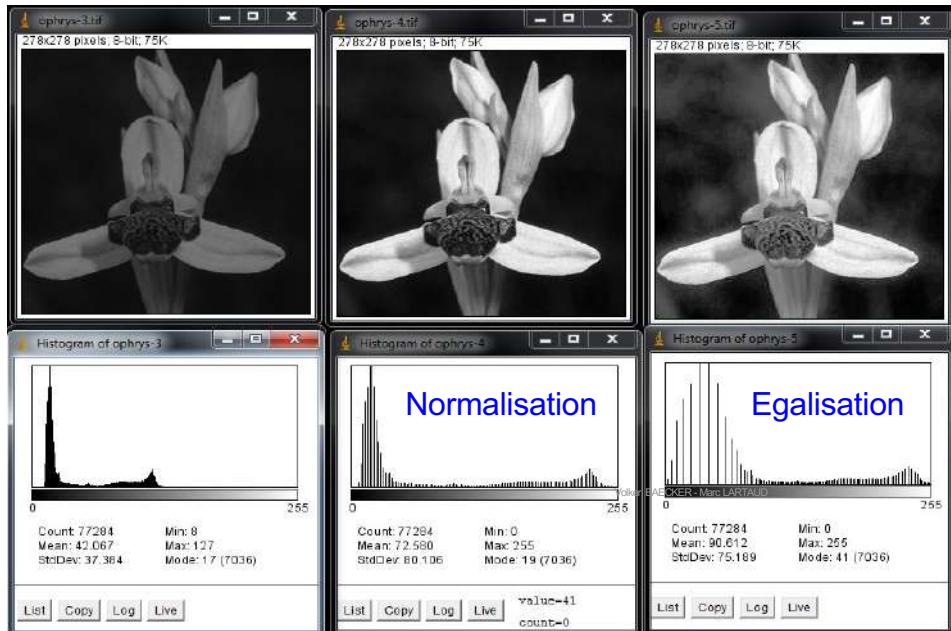


- Hue (*teinte, tonalité chromatique*) = perception de la couleur (bleu, vert, rouge, ...)
- Saturation (*saturation*) = vivacité des couleurs (couleurs grisâtres vs. vives)
- Lightness / Brightness (*luminosité / luminance*) = intensité lumineuse (aspect clair, foncé ou terne d'une couleur)

164

Correction non linéaire : Egalisation d'histogramme

Densité de probabilité normalisée pour "aplatisir" l'histogramme



165

ii) Egalisation d'histogramme

En pratique :

- calcul de l'histogramme $H(x)$ de l'image
- calcul de l'histogramme cumulé $H_c(x)$ correspondant :

$$H_c(x) = \sum_{i=0}^x H(i)$$

- construction d'une fonction de transfert :

$$LUT(x) = \frac{255}{M \times N} H_c(x)$$



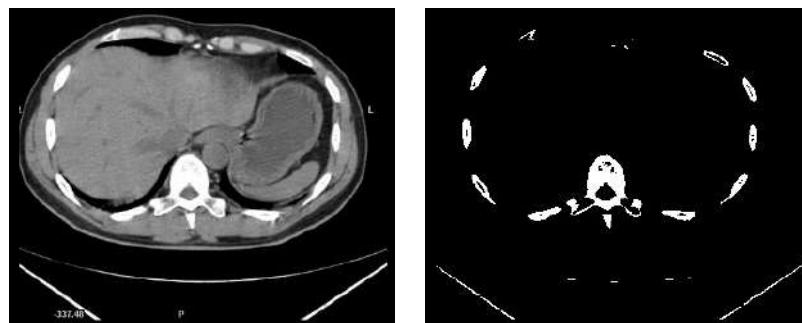
166

iii) Seuillage d'histogramme

Le seuillage binaire consiste à attribuer à chaque pixel la valeur 0 ou 1 en fonction de son intensité :

$$I_s(x, y) = \begin{cases} 0 & \text{si } I(x, y) \leq T \\ 1 & \text{si } I(x, y) > T \end{cases}$$

On appelle T le **seuil**, d'où le terme de seuillage

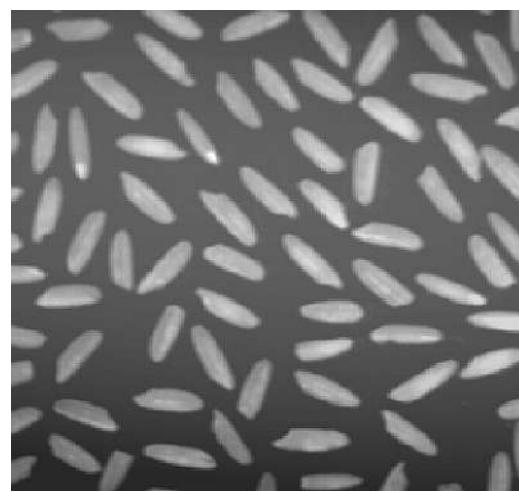


SORBONNE
UNIVERSITÉ

167

iii) Seuillage d'histogramme

Problème : trouver le bon seuil (si il existe)

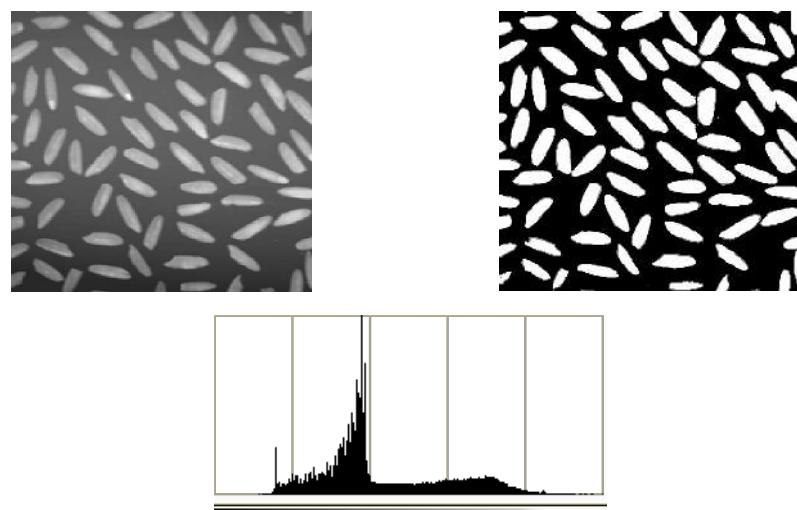


SORBONNE
UNIVERSITÉ

168

III) Seuillage d'histogramme

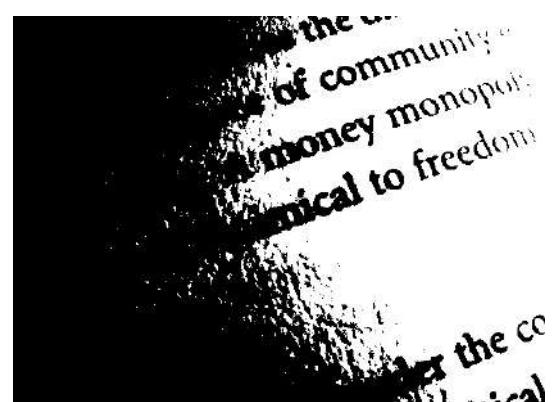
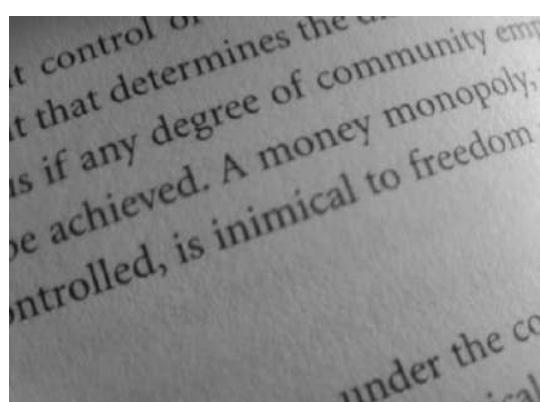
Utilisation de l'histogramme pour trouver le seuil



169

iii) Seuillage d'histogramme

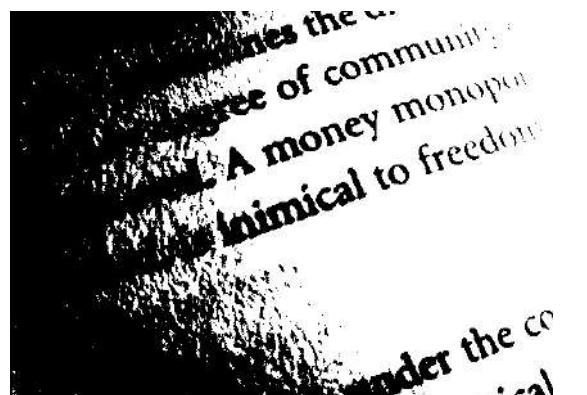
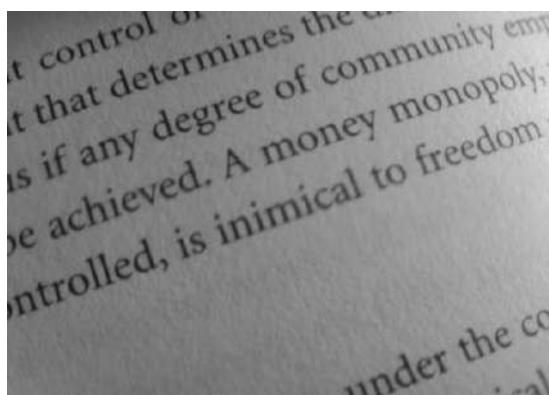
Seuillage simple : tests



170

iii) Seuillage d'histogramme

Seuillage simple : tests



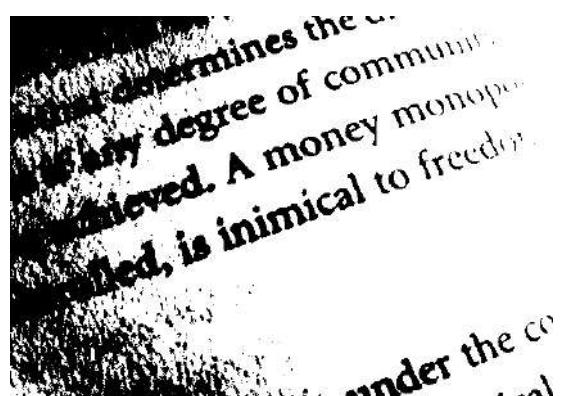
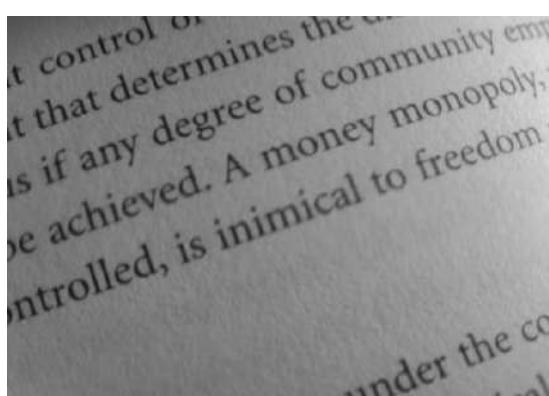
171



171

iii) Seuillage d'histogramme

Seuillage simple : tests



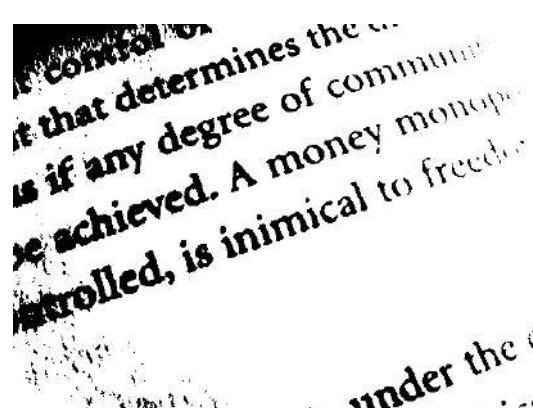
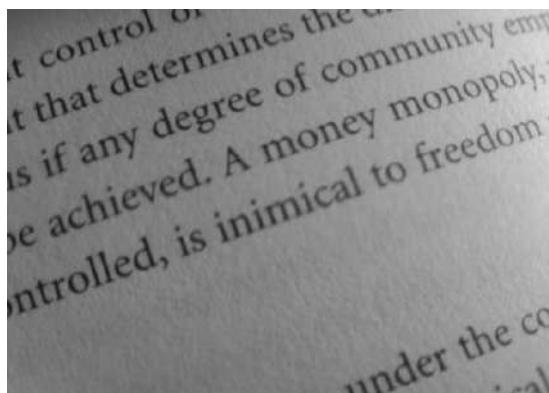
172



172

iii) Seuillage d'histogramme

Seuillage simple : tests



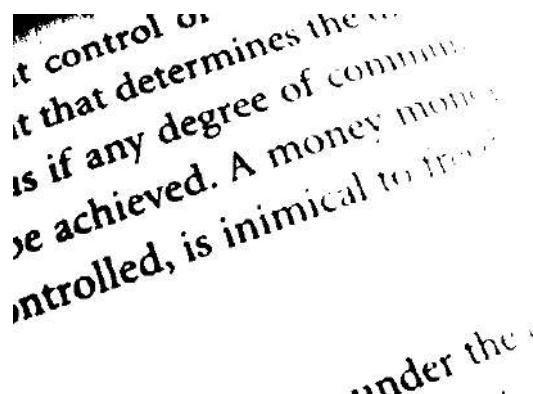
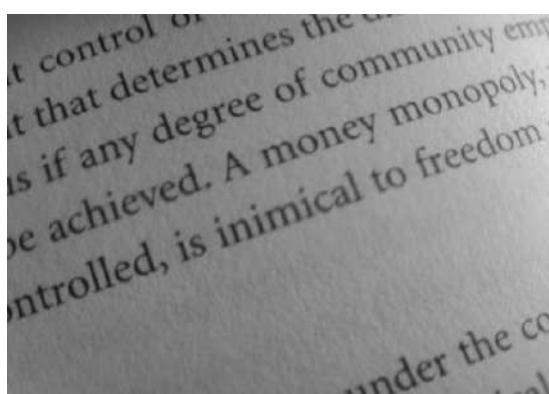
173



173

iii) Seuillage d'histogramme

Seuillage simple : tests



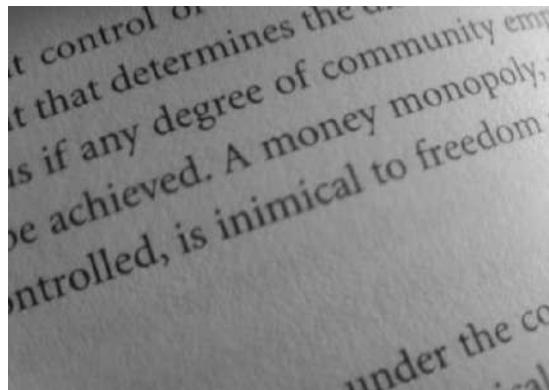
174



174

iii) Seuillage d'histogramme

Seuillage simple : tests



175



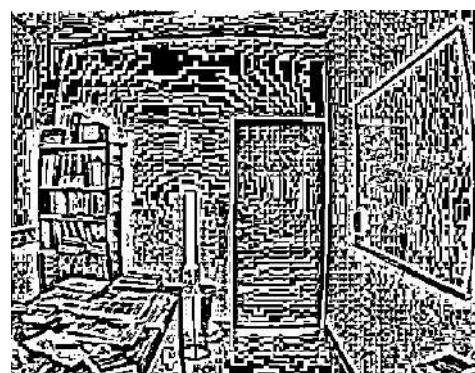
175

$$T = T[c(x,y), I(x,y)]$$

iii) Seuillage d'histogramme

Seuillage adaptatif : la valeur du seuil varie avec un critère $c(x,y)$ calculé dans le voisinage du pixel en question

Seuillage dynamique : la valeur du seuil varie avec la localisation du pixel



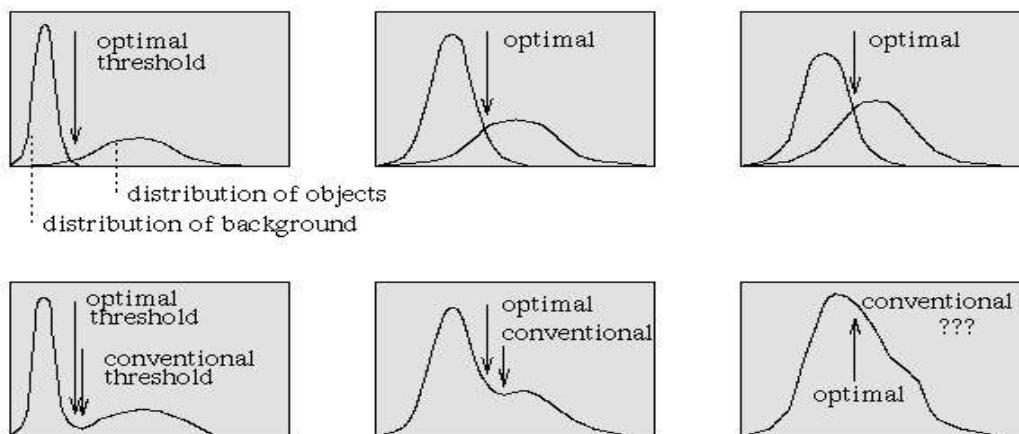
176

Dans cet exemple le seuil est la valeur moyenne d'une fenêtre 5×5 centrée en (x,y)



176

Seuillage adaptatif



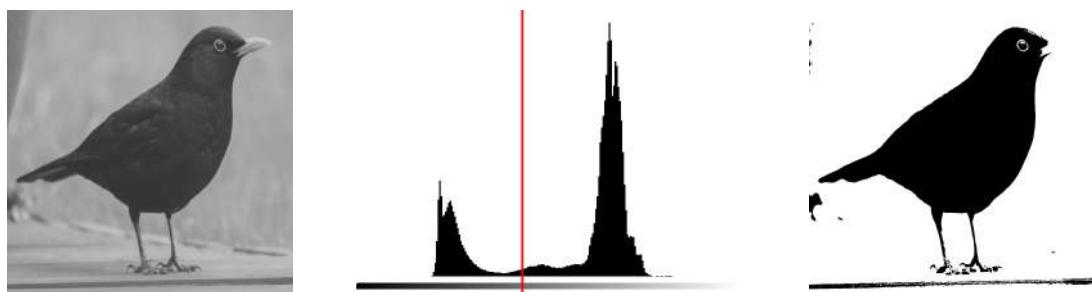
177



177

Seuillage adaptatif

Seuillage automatique : méthode d'Otsu



- choisir la coupe qui minimise la variance des 2 groupes

178

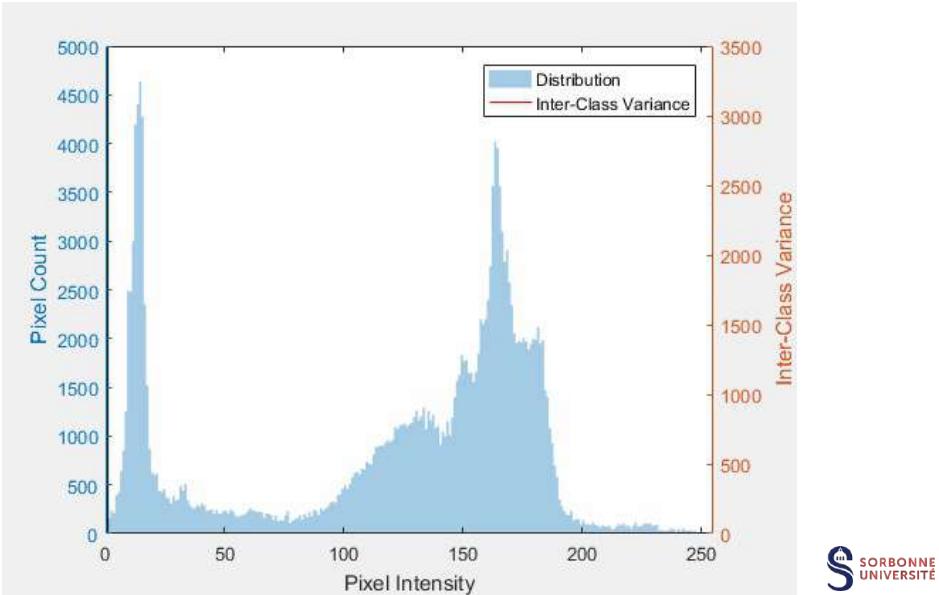


178

Seuillage adaptatif

Méthode d'Otsu

179

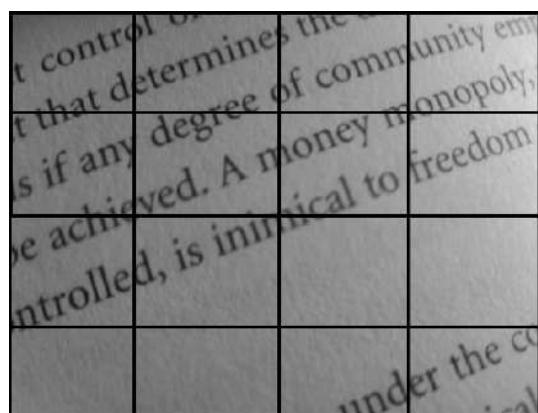


179

Seuillage adaptatif

Méthode d'Otsu

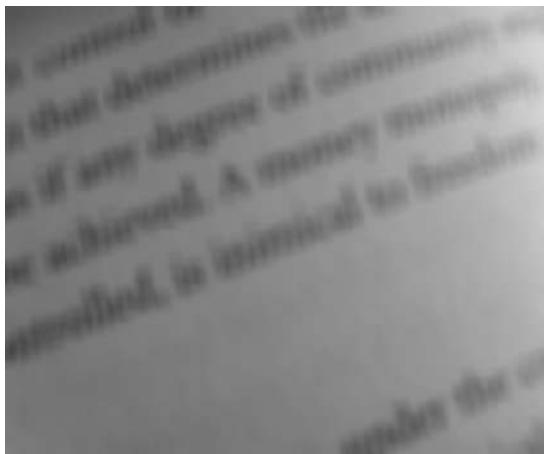
180



180

Seuillage adaptatif

Méthode utilisant du flou



181
seuillage du pixel selon que $I(x, y) > \text{blur}(I(x, y)) - \alpha$

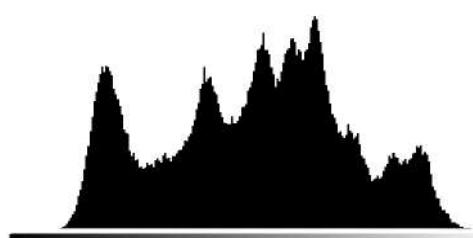
it control or
it that determines the
is if any degree of community emp
be achieved. A money monopoly
controlled, is inimical to freedom
under the co
ocial



181

Histogramme : invariance

Spatiale



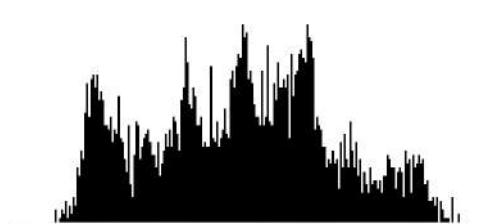
182



182

Histogramme : invariance

Définition 512×512 vs. 64×64

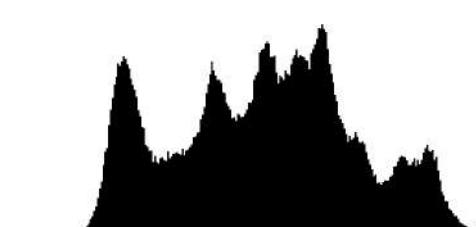
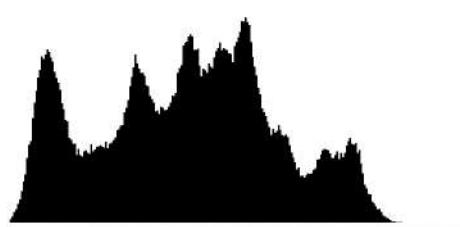


183

183

Histogramme : invariance

Luminance



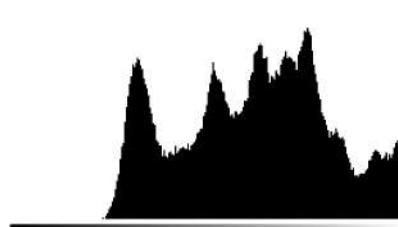
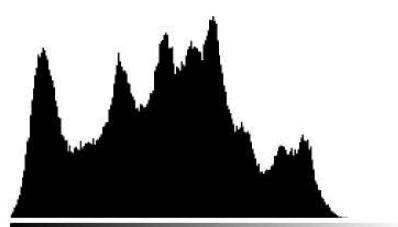
184

184

Histogramme

Saturation et clamp

Attention :
opération non réversible.



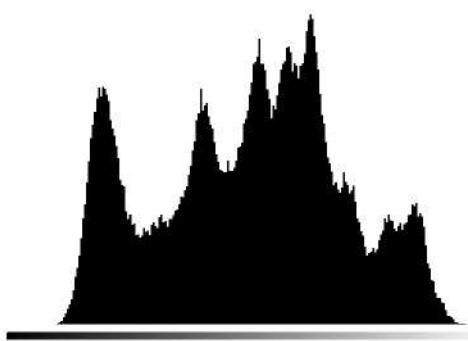
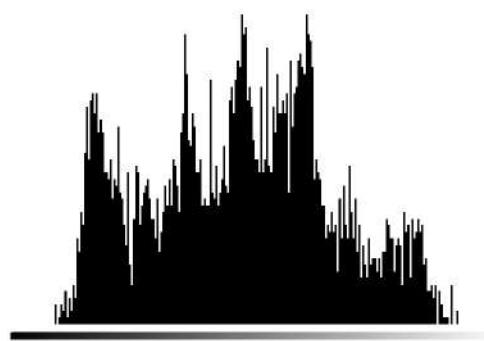
185

185

Distance d'histogrammes :

Question ouverte :

Peut-on estimer la ressemblance de 2 images en comparant leur histogramme?



186

186

b. Modèles linéaires

Traitements d'images



187

Filtrage

Le principe du filtrage est de modifier la valeur des pixels d'une image, généralement dans le but d'améliorer son aspect.

En pratique, il s'agit de créer une nouvelle image en se servant des valeurs des pixels de l'image d'origine.

Un filtre est une transformation mathématique (appelée **produit de convolution**) permettant de modifier la valeur d'un pixel en fonction des valeurs des pixels avoisinants, affectées de coefficients. Les calculs sont faits pour **chacune des trois composantes de couleur**.

Le filtre est représenté par un tableau (une matrice), caractérisé par ses dimensions et ses coefficients, dont le centre correspond au pixel concerné (cible).

En général, la somme des coefficients est égale à 1.

188



188

Modèle linéaire

Dans le modèle linéaire, la structure mathématique sous-jacente est l'**Espace Vectoriel**. Les opérateurs fondamentaux sont ceux qui préservent la structure d'espace vectoriel, c'est à dire les applications linéaires :

$$\begin{aligned} f(I+J) &= f(I) + f(J) \\ \lambda f(I) &= f(\lambda I) \end{aligned}$$

Pour les images, ces opérateurs correspondent aux **convolutions**.

189



189

Modèle linéaire : Convolution

Convolution : opérateur de base du traitement linéaire des images.

Apparue très tôt dans les premiers systèmes d'analyse d'images sous forme empirique et justifiée par des considérations d'implantation, ce n'est que plus tard qu'on a fourni des justifications physiques et fait le lien théorique avec les **filtres** et le **traitement du signal**.

190



190

Modèle linéaire : Convolution

1D (continue, discrète)

$$(f * g)(x) = \int_{\alpha=-\infty}^{\infty} f(\alpha)g(x-\alpha)d\alpha = \lim_{n \rightarrow \infty} \sum_{\alpha=-n}^n f(\alpha)g(x-\alpha)$$

2D (continue, discrète)

$$\begin{aligned} (f * g)(x) &= \int_{\alpha=-\infty}^{\infty} \int_{\beta=-\infty}^{\infty} f(\alpha, \beta)g(x-\alpha, y-\beta) d\alpha d\beta \\ &= \lim_{n \rightarrow \infty} \sum_{\alpha=-n}^n \sum_{\beta=-n}^n f(\alpha, \beta)g(x-\alpha, y-\beta) \end{aligned}$$

191



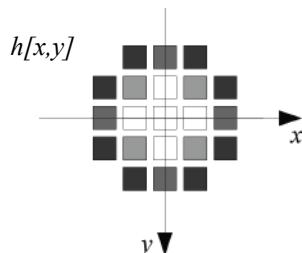
191

Modèle linéaire : Convolution

Soit I une image numérique.

Soit h une fonction de $[x_1, x_2] \times [y_1, y_2]$ à valeurs réelles.

La convolution de I par h est définie par :



$$(I * h)[x, y] = \sum_{i=x_1}^{x_2} \sum_{j=y_1}^{y_2} h[i, j] \cdot I[x - i, y - j]$$

La fonction h est appelée **noyau de convolution**

Les nouvelles valeurs des pixels de I sont calculées par **produit scalaire** entre le noyau et le **voisinage** correspondant du pixel .

Importance de la norme du noyau

Conditions aux bords de l'image

192



192

Propriétés de la convolution

Commutativité

$$h * g = g * h$$

Associativité

$$(h * g) * k = h * (g * k) = h * g * k$$

Distributivité

$$h * (g + k) = (h * g) + (h * k)$$

Elément neutre

$$\delta = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \rightarrow (\delta * g)(x, y) = g(x, y)$$

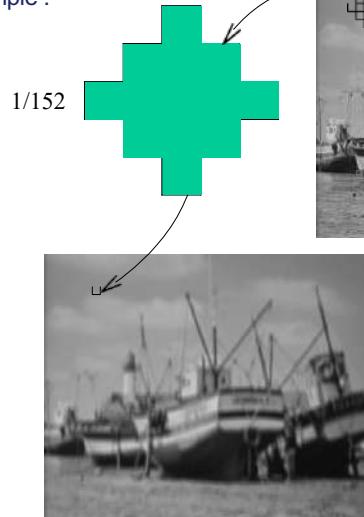
193



193

Modèle linéaire : Convolution

Exemple :



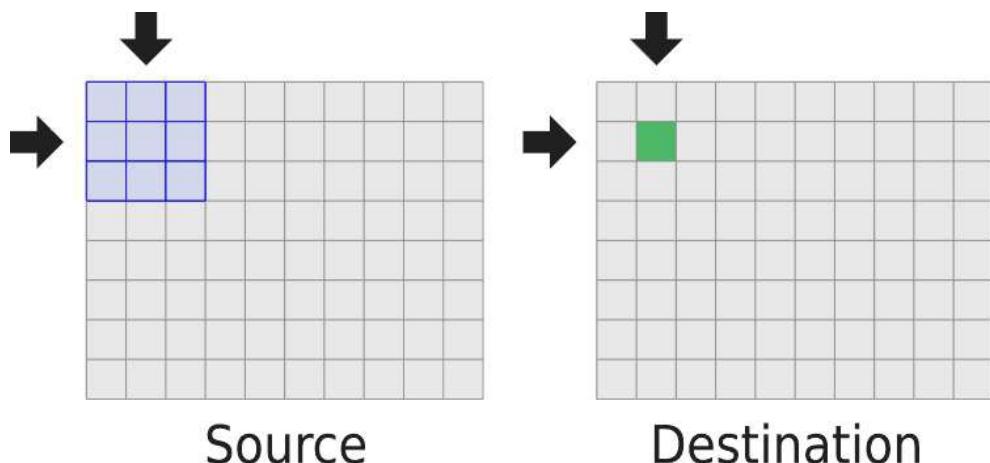
Pour calculer une convolution, on remplace la valeur de chaque pixel par la valeur du produit scalaire entre les valeurs du **noyau de convolution** et les **valeurs du voisinage** du pixel considéré (par rapport à l'origine (0,0) du noyau de convolution).

Implémentation « parallèle » fortement encouragée !



194

Filtres de convolution

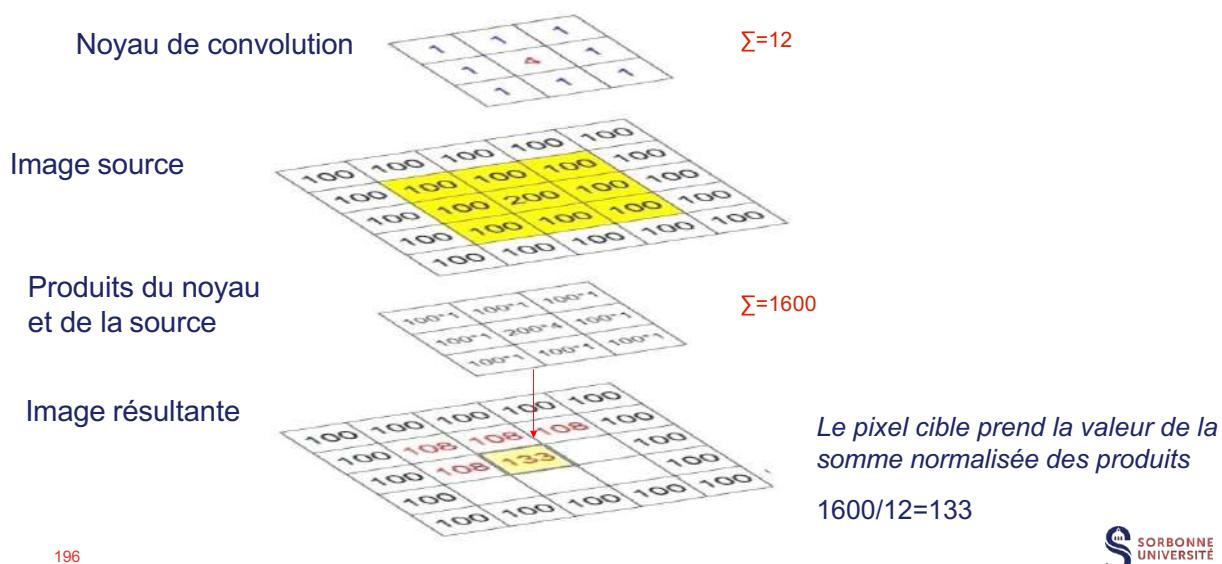


195



195

Filtres de convolution



196



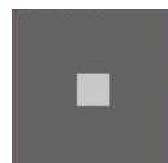
196

Modèle linéaire : Convolution

100	100	100	100	100
100	100	100	100	100
100	100	200	100	100
100	100	100	100	100
100	100	100	100	100

$$\begin{matrix} * & \begin{matrix} 1 & 1 & 1 \\ 1 & 4 & 1 \\ 1 & 1 & 1 \end{matrix} & = \end{matrix}$$

100	100	100	100	100
100				100
100				100
100				100
100	100	100	100	100



197

197

Filtres de convolution

100	100	100	100	100
100	100	100	100	100
100	100	200	100	100
100	100	100	100	100
100	100	100	100	100

$$\begin{matrix} * & \begin{matrix} 1 & 1 & 1 \\ 1 & 4 & 1 \\ 1 & 1 & 1 \end{matrix} & = \end{matrix}$$

100	100	100	100	100
100		108		100
100				100
100				100
100	100	100	100	100

100*1	100*1	100*1
100*1	100*4	100*1
100*1	100*1	200*1

Somme / 12 = 108

NB : dans ce cas précis, 12 = somme des pondérations du noyau de convolution
Opération de normalisation

198

Filtres de convolution

100	100	100	100	100
100	100	100	100	100
100	100	200	100	100
100	100	100	100	100
100	100	100	100	100

$$\begin{matrix} & 1 & 1 & 1 \\ * & 1 & 4 & 1 \\ & 1 & 1 & 1 \end{matrix}$$

100	100	100	100	100
100	108	108	108	100
100				100
100				100
100	100	100	100	100

100*1	100*1	100*1
100*1	100*4	100*1
100*1	200*1	100*1

Somme / 12 = 108

199



199

Filtres de convolution

100	100	100	100	100
100	100	100	100	100
100	100	200	100	100
100	100	100	100	100
100	100	100	100	100

$$\begin{matrix} & 1 & 1 & 1 \\ * & 1 & 4 & 1 \\ & 1 & 1 & 1 \end{matrix}$$

100	100	100	100	100
100	108	108	108	100
100				100
100				100
100	100	100	100	100

100*1	100*1	100*1
100*1	100*4	100*1
200*1	100*1	100*1

Somme / 12 = 108

200



200

Filtres de convolution

$$\begin{array}{|c|c|c|c|c|} \hline
 100 & 100 & 100 & 100 & 100 \\ \hline
 100 & 100 & 100 & 100 & 100 \\ \hline
 100 & 100 & 200 & 100 & 100 \\ \hline
 100 & 100 & 100 & 100 & 100 \\ \hline
 100 & 100 & 100 & 100 & 100 \\ \hline
 \end{array} * \begin{array}{|c|c|c|} \hline
 1 & 1 & 1 \\ \hline
 1 & 4 & 1 \\ \hline
 1 & 1 & 1 \\ \hline
 \end{array} = \begin{array}{|c|c|c|c|c|} \hline
 100 & 100 & 100 & 100 & 100 \\ \hline
 100 & 108 & 108 & 108 & 100 \\ \hline
 100 & 108 & & & 100 \\ \hline
 100 & & & & 100 \\ \hline
 100 & 100 & 100 & 100 & 100 \\ \hline
 \end{array}$$

$$\begin{array}{|c|c|c|} \hline
 100^*1 & 100^*1 & 100^*1 \\ \hline
 100^*1 & 100^*4 & 200^*1 \\ \hline
 100^*1 & 100^*1 & 100^*1 \\ \hline
 \end{array} \text{ Somme / 12 = } 108$$

201



201

Filtres de convolution

$$\begin{array}{|c|c|c|c|c|} \hline
 100 & 100 & 100 & 100 & 100 \\ \hline
 100 & 100 & 100 & 100 & 100 \\ \hline
 100 & 100 & 200 & 100 & 100 \\ \hline
 100 & 100 & 100 & 100 & 100 \\ \hline
 100 & 100 & 100 & 100 & 100 \\ \hline
 \end{array} * \begin{array}{|c|c|c|} \hline
 1 & 1 & 1 \\ \hline
 1 & 4 & 1 \\ \hline
 1 & 1 & 1 \\ \hline
 \end{array} = \begin{array}{|c|c|c|c|c|} \hline
 100 & 100 & 100 & 100 & 100 \\ \hline
 100 & 108 & 108 & 108 & 100 \\ \hline
 100 & 108 & 133 & & 100 \\ \hline
 100 & & & & 100 \\ \hline
 100 & 100 & 100 & 100 & 100 \\ \hline
 \end{array}$$

$$\begin{array}{|c|c|c|} \hline
 100^*1 & 100^*1 & 100^*1 \\ \hline
 100^*1 & 200^*4 & 100^*1 \\ \hline
 100^*1 & 100^*1 & 100^*1 \\ \hline
 \end{array} \text{ Somme / 12 = } 133$$

202

NB : dans ce cas précis, 12 = somme des pondérations du noyau de convolution
Opération de normalisation

202

Filtres de convolution

100	100	100	100	100
100	100	100	100	100
100	100	200	100	100
100	100	100	100	100
100	100	100	100	100

$$\begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 4 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}$$

100	100	100	100	100
100	108	108	108	100
100	108	133	108	100
100				100
100	100	100	100	100

100*1	100*1	100*1
200*1	100*4	100*1
100*1	100*1	100*1

Somme / 12 = 108

203



203

Filtres de convolution

100	100	100	100	100
100	100	100	100	100
100	100	200	100	100
100	100	100	100	100
100	100	100	100	100

$$\begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 4 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}$$

100	100	100	100	100
100	108	108	108	100
100	108	133	108	100
100	108			100
100	100	100	100	100

100*1	100*1	200*1
100*1	100*4	100*1
100*1	100*1	100*1

Somme / 12 = 108

204



204

Filtres de convolution

100	100	100	100	100
100	100	100	100	100
100	100	200	100	100
100	100	100	100	100
100	100	100	100	100

$$\begin{matrix} & 1 & 1 & 1 \\ * & 1 & 4 & 1 \\ & 1 & 1 & 1 \end{matrix}$$

100	100	100	100	100
100	108	108	108	100
100	108	133	108	100
100	108	108		100
100	100	100	100	100

100*1	200*1	100*1
100*1	100*4	100*1
100*1	100*1	100*1

Somme / 12 = 108

205



205

Filtres de convolution

100	100	100	100	100
100	100	100	100	100
100	100	200	100	100
100	100	100	100	100
100	100	100	100	100

$$\begin{matrix} & 1 & 1 & 1 \\ * & 1 & 4 & 1 \\ & 1 & 1 & 1 \end{matrix}$$

100	100	100	100	100
100	108	108	108	100
100	108	133	108	100
100	108	108	108	100
100	100	100	100	100

200*1	100*1	100*1
100*1	100*4	100*1
100*1	100*1	100*1

Somme / 12 = 108

206

NB : dans ce cas précis, 12 = somme des pondérations du noyau de convolution
Opération de normalisation

206

Filtres de convolution

$$\begin{array}{|c|c|c|c|c|} \hline
 100 & 100 & 100 & 100 & 100 \\ \hline
 100 & 100 & 100 & 100 & 100 \\ \hline
 100 & 100 & 200 & 100 & 100 \\ \hline
 100 & 100 & 100 & 100 & 100 \\ \hline
 100 & 100 & 100 & 100 & 100 \\ \hline
 \end{array} * \begin{array}{|c|c|c|} \hline
 1 & 1 & 1 \\ \hline
 1 & 4 & 1 \\ \hline
 1 & 1 & 1 \\ \hline
 \end{array} = \begin{array}{|c|c|c|c|c|} \hline
 100 & 100 & 100 & 100 & 100 \\ \hline
 100 & \textcolor{red}{108} & \textcolor{red}{108} & \textcolor{red}{108} & 100 \\ \hline
 100 & \textcolor{red}{108} & \textcolor{red}{133} & \textcolor{red}{108} & 100 \\ \hline
 100 & \textcolor{red}{108} & \textcolor{red}{108} & \textcolor{red}{108} & 100 \\ \hline
 100 & 100 & 100 & 100 & 100 \\ \hline
 \end{array}$$



207

Le résultat d'un tel filtrage est un lissage de l'image : filtre passe bas



207

Modèle linéaire : Convolution

A vous de jouer !

$$\begin{array}{|c|c|c|c|c|} \hline
 100 & 100 & 100 & 100 & 100 \\ \hline
 100 & 100 & 100 & 100 & 100 \\ \hline
 100 & 100 & \textcolor{yellow}{150} & 100 & 100 \\ \hline
 100 & 100 & 100 & 100 & 100 \\ \hline
 100 & 100 & 100 & 100 & 100 \\ \hline
 \end{array} * \begin{array}{|c|c|c|} \hline
 0 & -1 & 0 \\ \hline
 -1 & 5 & -1 \\ \hline
 0 & -1 & 0 \\ \hline
 \end{array} \rightarrow$$

208



208

Coins et bords ...

Le noyau est bidimensionnel. Par consequent, on rencontre des problèmes lorsque le noyau est proche des bords ou des coins.

Habituellement, on crée des pixels supplémentaires près des bords.

Il existe plusieurs façons de créer des pixels supplémentaires:

- Définir une valeur constante pour ces pixels
- Dupliquer les pixels de bord
- Refléter les bords (effet miroir)
- Déformer l'image autour (copier les pixels de l'autre extrémité)

Cela résout généralement les problèmes qui peuvent survenir.

209

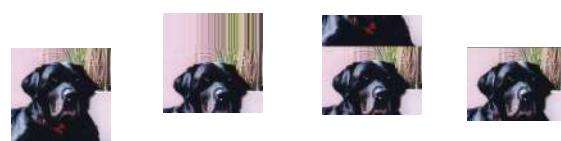


209

Modèle linéaire : Convolution

- Gestion des effets de bords:

- Pixels à 0 ou 255;
- Pixels identiques;
- Miroir;
- Image périodique;
- Ignorer $m/2$ pixels sur le bord.



Exemple sur le
bord haut

- La dimension du masque définit la localité de l'effet de la convolution.

211



211

Filtrage : lissage

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9



212



212

Filtrage : accentuation

0	-0.5	0
-0.5	3	-0.5
0	-0.5	0



213



213

Filtrage : gradient (Sobel)

-1	0	1
-2	0	2
-1	0	1



214



214

Filtrage : filtre médian

Considérons neuf pixels en niveaux de gris, dont une valeur est aberrante (ici 255) :

2	4	12
2	255	3
7	9	3

Le filtre médian va d'abord trier ces valeurs par ordre croissant :

2, 2, 3, 3, 4, 7, 9, 12, 255

et prendre la valeur médiane (la cinquième valeur), ici la valeur 4. La sortie du filtre donnera :

2	4	12
2	4	3
7	9	3

215



215

Filtrage : filtre médian



Image bruitée



Image filtrée 1 fois



Image filtrée 2 fois

216

216

Modèle linéaire : Convolution

La convolution dans le traitement d'images :

- Filtrage linéaire des bruits
- Détection de contours
- Rehaussement de contraste

217

217

c. Modèles fréquentiels



Traitement d'Images

218

Modèle fréquentiel

- Basses fréquences
 - ✓ Zones homogènes, continues
- Hautes fréquences
 - ✓ Détails, contours



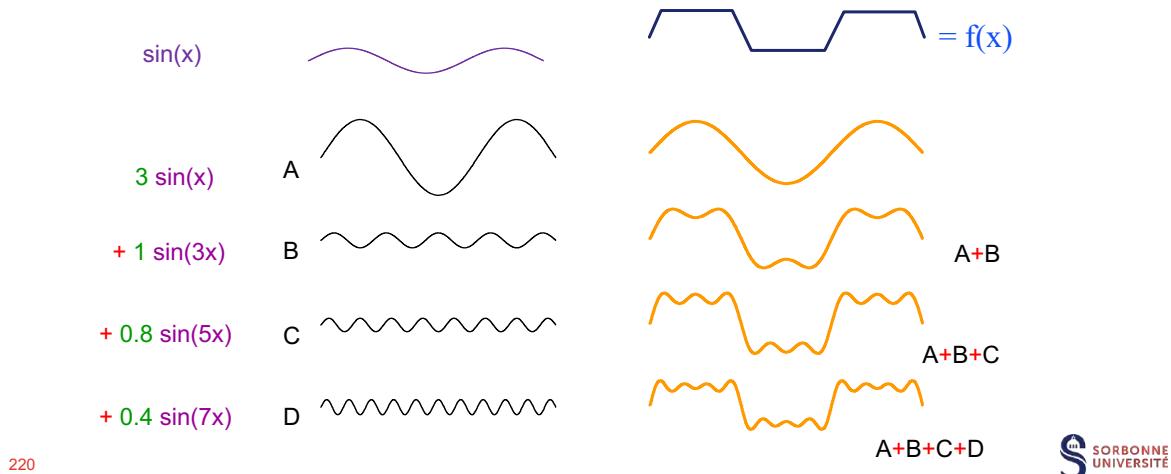
219

219

103

Modèle fréquentiel

Le modèle fréquentiel tend à décrire l'image en termes de *structures périodiques*, en la décomposant *dans une base de fonctions périodiques simples*, comme des sinusoïdes

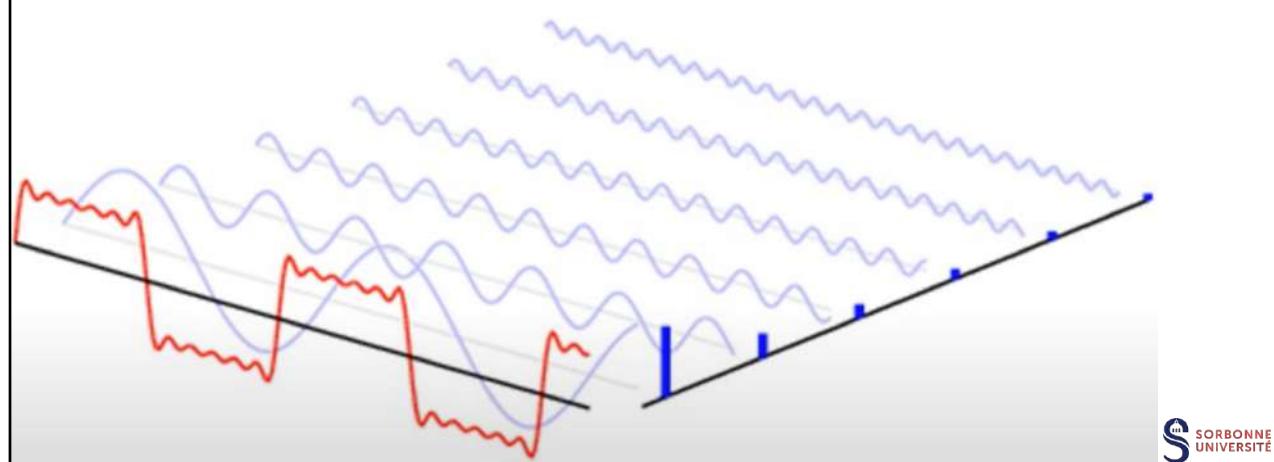


220



Modèle fréquentiel pour le traitement d'images

Principe : Toute fonction, même non périodique, mais dont l'aire sous la courbe est finie, peut être représentée par l'intégrale de fonctions sinus et cosinus, chacune multipliée par un coefficient différent (séries de Fourier)



222



Modèle fréquentiel pour le traitement d'images

Principe : Toute fonction, même non périodique, mais dont l'aire sous la courbe est finie, peut être représentée par l'intégrale de fonctions sinus et cosinus, chacune multipliée par un coefficient différent

Analyse de Fourier :

Toute fonction périodique peut être décomposée en une somme de sinus et cosinus

Toute fonction périodique peut être décomposée en séries de Fourier

$$f(x) = a_0 + \sum_{n=1}^{\infty} (a_n \cos nx + b_n \sin nx)$$



223

Modèle fréquentiel pour le traitement d'images

Analyse de Fourier :

$$f(x) = a_0 + \sum_{n=1}^{\infty} (a_n \cos nx + b_n \sin nx)$$

Coefficients de Fourier :

$$a_0 = \frac{2}{T} \int_0^T f(x) dx$$

$$a_n = \frac{2}{T} \int_0^T f(x) \cos nx dx$$

$$b_n = \frac{2}{T} \int_0^T f(x) \sin nx dx$$

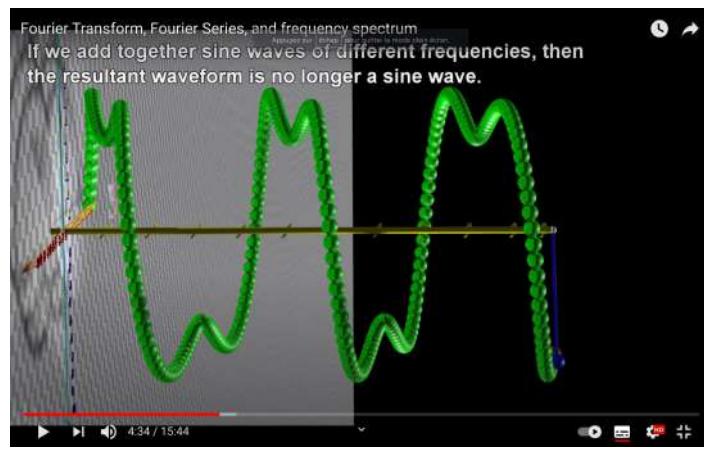


224

Modèle fréquentiel pour le traitement d'images

Principe : Toute fonction, même non périodique, mais dont l'aire sous la courbe est finie, peut être représentée pas l'intégrale de fonctions sinus et cosinus, chacune multipliée par un coefficient différent

<https://www.youtube.com/watch?v=r18Gi8lSkfM&t=36s>



225

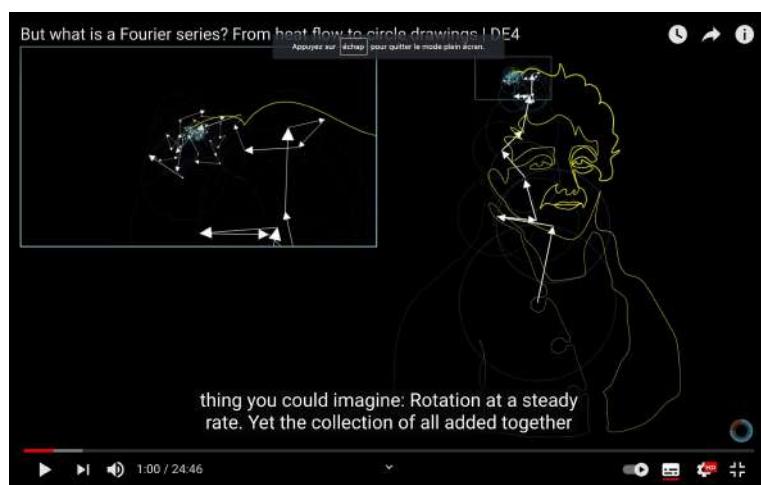


225

Modèle fréquentiel pour le traitement d'images

Principe : Toute fonction, même non périodique, mais dont l'aire sous la courbe est finie, peut être représentée pas l'intégrale de fonctions sinus et cosinus, chacune multipliée par un coefficient différent

<https://www.youtube.com/watch?v=r6sGWTCMz2k>



226



226

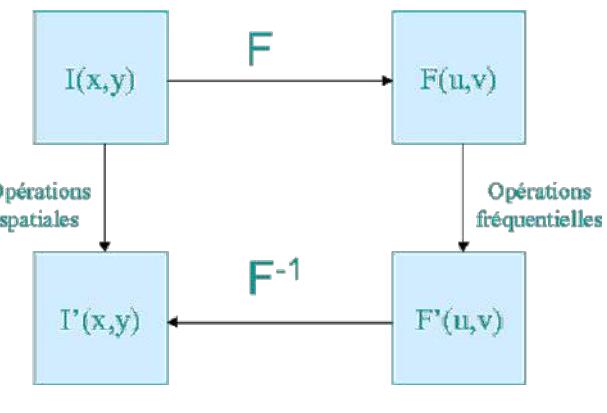
Modèle fréquentiel pour le traitement d'images

Principe : Toute fonction, même non périodique, mais dont l'aire sous la courbe est finie, peut être représentée par l'intégrale de fonctions sinus et cosinus, chacune multipliée par un coefficient différent.

Application :

- Convertir du domaine spatial vers le domaine fréquentiel,
- Effectuer des manipulations directement dans le domaine fréquentiel,
- Transformée inverse de la solution du domaine fréquentiel au domaine spatial.

Sans perte d'information !

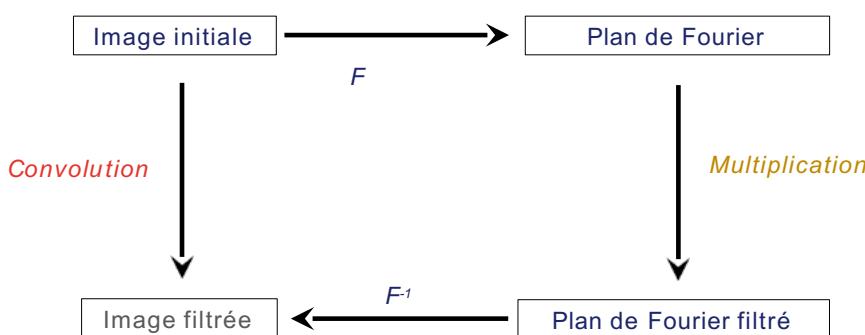


227



227

Filtrage (spatial) - masquage (fréquentiel)



Dans le **domaine spatial**, le filtrage se fait par **convolution**.

Dans le **domaine spectral**, il se fait par **multiplication** (ou **masquage** de l'image).

228



228

Transformée de Fourier

Outil fondamental d'analyse en traitement du signal, le pendant bidimensionnel de la **Transformée de Fourier (TF)** et sa version discrète peuvent être appliquées avec profit aux images numériques.

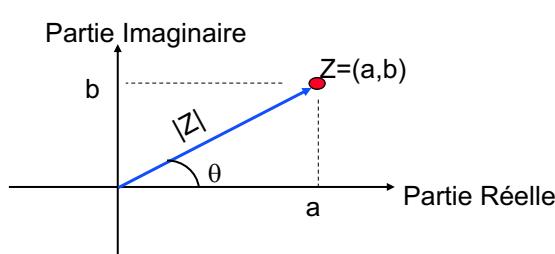
Si l'utilisation de la TF en tant qu'outil analytique et algorithmique s'est estompée en traitement d'images au profit d'approches mieux adaptées à la **localisation spatiale des fréquences (ondelettes)**, elle reste un **outil théorique et pédagogique important** : la formalisation du phénomène de l'aliasing et des contraintes d'échantillonnage en est un exemple.

229



229

Modèle fréquentiel : rappel sur les nombres complexes



$$Z = \operatorname{Re}(Z) + i \operatorname{Im}(Z)$$

$$= a + ib$$

$$= |Z| e^{i\theta}$$

$$i = \sqrt{-1} \quad (i^2 = -1)$$

$$e^{i\theta} = \cos \theta + i \sin \theta \quad (\text{vecteur tournant unité})$$

$$|Z| = \sqrt{a^2 + b^2} \quad (\text{Norme; Amplitude de Fourier})$$

$$\theta = \operatorname{tg}^{-1}(b/a) \quad (\text{phase})$$

$$\bar{Z} = Z^* = a - ib = |Z| e^{-i\theta} \quad (\text{complexe conjugué})$$

230



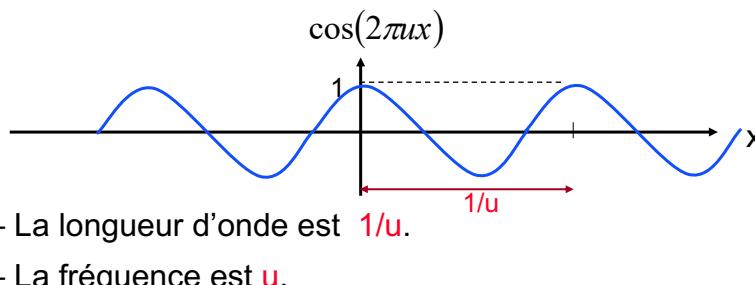
230

Modèle fréquentiel : rappel sur les nombres complexes

$e^{2\pi iux}$ = fonctions de bases 1D

$$\operatorname{Re}(e^{2\pi iux}) = \cos(2\pi ux)$$

$$\operatorname{Im}(e^{2\pi iux}) = \sin(2\pi ux)$$



- La longueur d'onde est $1/u$.
- La fréquence est u .

231



231

Transformée de Fourier

La transformée de Fourier permet la décomposition d'un signal f en **combinaison linéaire de sinusoïdes complexes**, dont les coefficients $F(u,v)$ dits **coefficients de Fourier**, fournissent des informations sur les fréquences (u,v) et permettent des manipulations dans le domaine fréquentiel. $F(u,v)$ porte le nom de **spectre** du signal f .

Transformée de Fourier bidimensionnelle dans le domaine continu :

- Directe $F(u,v) = \int_{x=-\infty}^{\infty} \int_{y=-\infty}^{\infty} f(x,y) e^{-2i\pi(ux+vy)} dx dy$

(x,y) sont les coordonnées du domaine spatial

- Inverse $f(x,y) = \int_{u=-\infty}^{\infty} \int_{v=-\infty}^{\infty} F(u,v) e^{2i\pi(ux+vy)} du dv$

(u,v) sont les coordonnées du domaine fréquentiel

232

$$e^{i\theta} = \cos \theta + i \sin \theta, \quad i = \sqrt{-1}$$



232

Modèle fréquentiel : transformée de Fourier

La transformée de Fourier permet la décomposition d'un signal f en **combinaison linéaire de sinusoïdes complexes**, dont les coefficients $F[u,v]$ dit **coefficients de Fourier**, fournissent des informations sur les fréquences (u,v) et permettent des manipulations dans le domaine fréquentiel. $F(u,v)$ porte le nom de **spectre** du signal f .

Transformée de Fourier **discrète bidimensionnelle** :

- Directe
- Inverse

$$F(u,v) = \frac{1}{NM} \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} f(x,y) e^{-2\pi i \left(\frac{ux}{N} + \frac{vy}{M}\right)} \quad (x,y) = \text{coord. domaine spatial}$$

$$f(x,y) = \sum_{u=0}^{N-1} \sum_{v=0}^{M-1} F(u,v) e^{+2\pi i \left(\frac{ux}{N} + \frac{vy}{M}\right)} \quad (x,y) = \text{coord. domaine fréquentiel}$$

234

$$e^{i\theta} = \cos \theta + i \sin \theta, \quad i = \sqrt{-1}$$



234

Modèle fréquentiel : transformée de Fourier

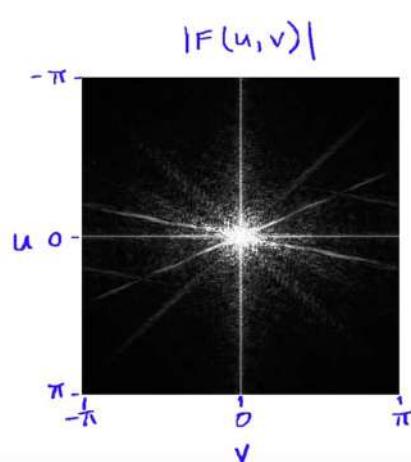
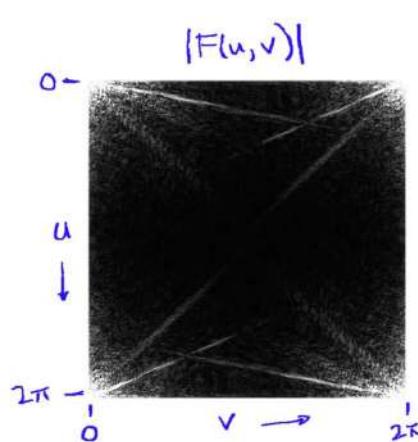


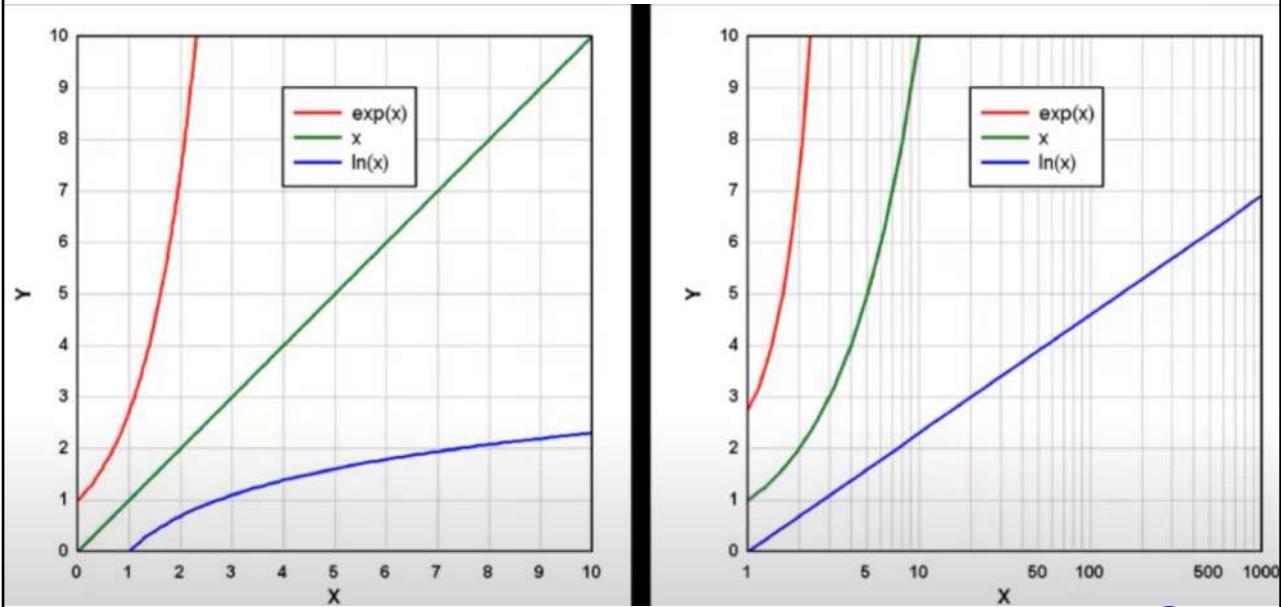
Diagramme d'amplitudes
Centrée en $(u,v)=(0,0)$ (basses fréquences)

235



235

Similarité – transformée logarithmique



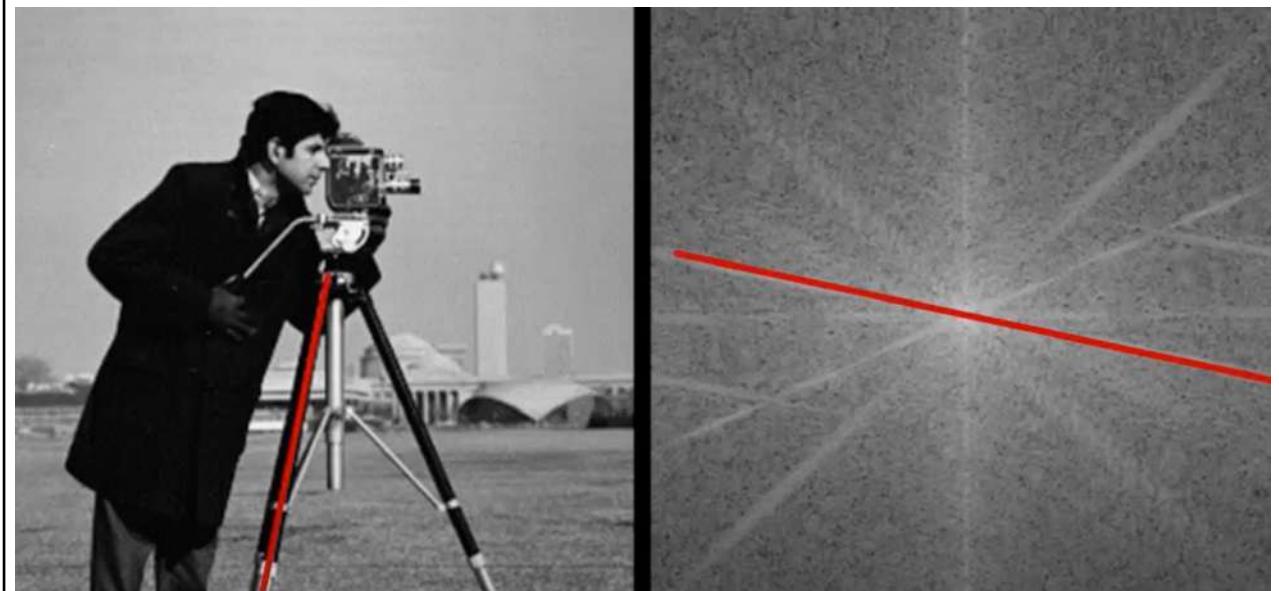
236

Modèle fréquentiel : transformée de Fourier



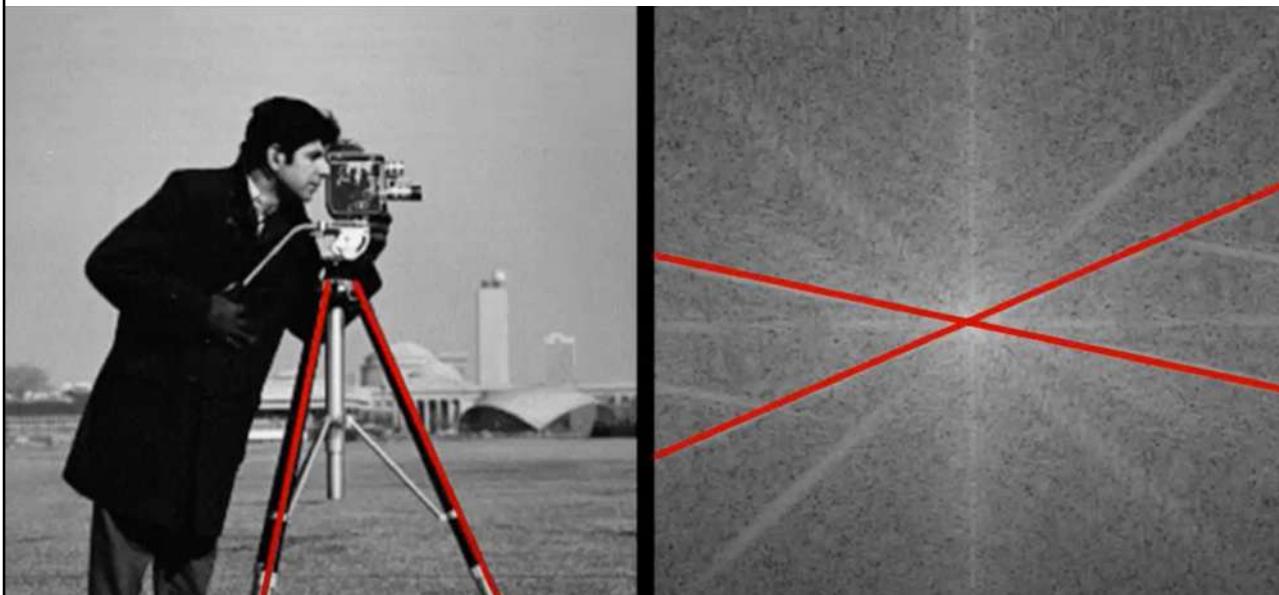
237

Modèle fréquentiel : transformée de Fourier



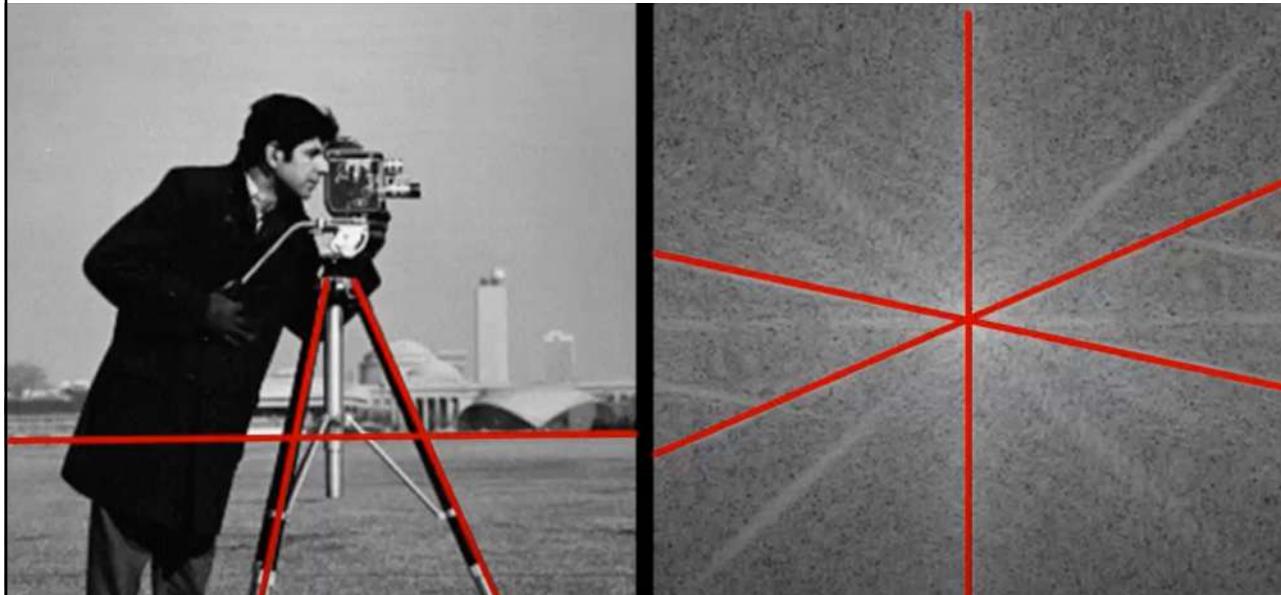
238

Modèle fréquentiel : transformée de Fourier



239

Modèle fréquentiel : transformée de Fourier



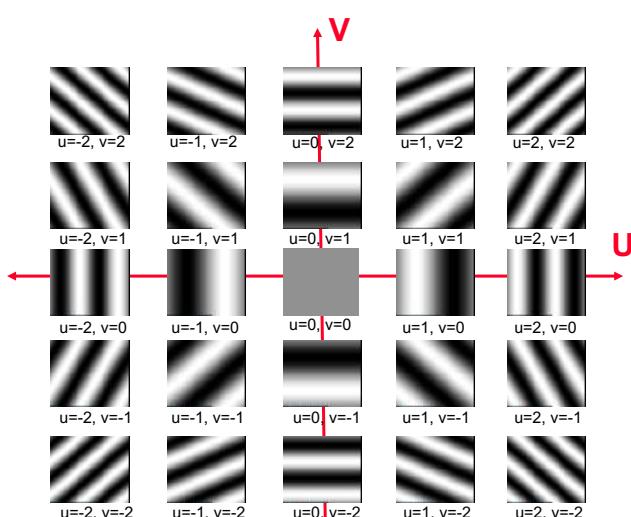
240

Modèle fréquentiel : transformée de Fourier

Fonctions de base :

$$e^{2\pi i(ux+vy)}$$

Tableau de correspondance entre les paramètres (u, v) et le signal modélisé :



La fréquence est $\sqrt{u^2 + v^2}$. La direction est u/v .

241

241

Modèle fréquentiel : transformée de Fourier

Représentation transformée 2D : image à valeurs complexes

Décomposition du spectre en images visibles :

- Spectre d'amplitude
- Spectre de phase

Pour l'affichage on peut utiliser $\log(1 + |F(u,v)|)$

Importance des deux composantes

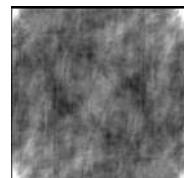


Image reconstruite à l'aide
de l'amplitude uniquement

Image reconstruite à l'aide
de la phase uniquement

242

242

Modèle fréquentiel : transformée de Fourier

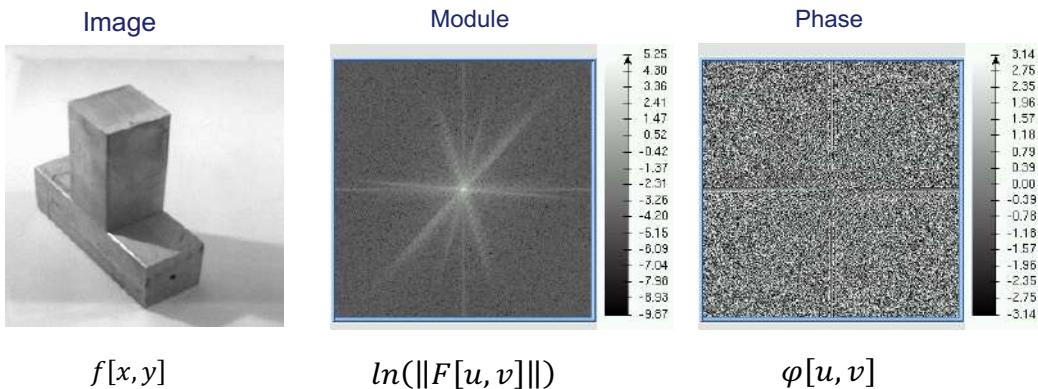
Phase d'une image plus amplitude de l'autre



243

243

Modèle fréquentiel : transformée de Fourier



244



244

Propriété de la transformée de Fourier

Ecriture sous forme module / phase : $F[u, v] = \|F[u, v]\|e^{-i\varphi[u, v]}$

Périodicité : $F[u, v] = F[u+w, v+h]$

Symétrie : si F est la transformée de Fourier d'une fonction réelle f :

$$F[u, v] = \overline{F[-u, -v]} \quad \text{donc} \quad \|F[u, v]\| = \|F[-u, -v]\| \quad \text{et} \quad \varphi[u, v] = -\varphi[-u, -v]$$

Correspondance convolution-produit : $f_1[x, y] * f_2[x, y] \rightarrow F_1[u, v] \cdot F_2[u, v]$

$$f_1[x, y] \cdot f_2[x, y] \rightarrow F_1[u, v] * F_2[u, v]$$

Dérivation :

$$\frac{\delta f[x, y]}{\delta x} \rightarrow iuF[u, v]$$

$$\frac{\delta f[x, y]}{\delta y} \rightarrow ivF[u, v]$$

245



245

Propriété de la transformée de Fourier

Linéarité : $a \cdot f_1[x, y] + b \cdot f_2[x, y] \rightarrow a \cdot F_1[u, v] + b \cdot F_2[u, v]$

Translation spatiales / fréquentielles :

$$f[x - x', y - y'] \rightarrow F[u, v] \cdot e^{-2i\pi(ux' + vy')/wh}$$

$$f[x, y] \cdot e^{-2i\pi(u'x + v'y)/wh} \rightarrow F[u - u', v - v']$$

Théorème de Parseval :

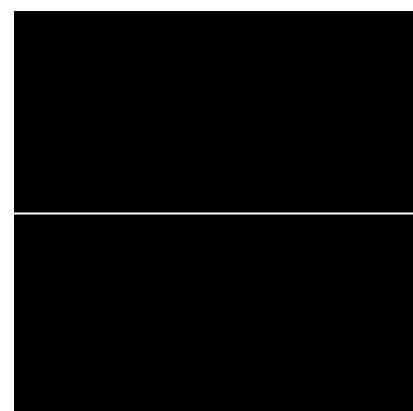
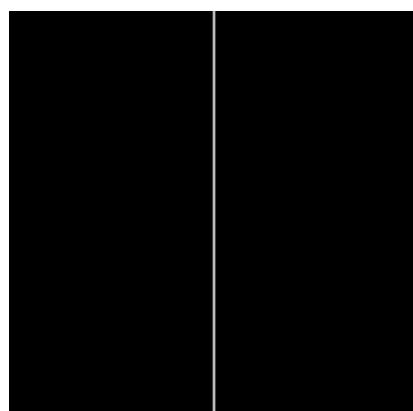
$$\sum_{x=0}^{w-1} \sum_{y=0}^{h-1} \|f[x, y]\|^2 = \frac{1}{wh} \sum_{u=0}^{w-1} \sum_{v=0}^{h-1} \|F[u, v]\|^2$$

246



246

Modèle fréquentiel : transformée de Fourier

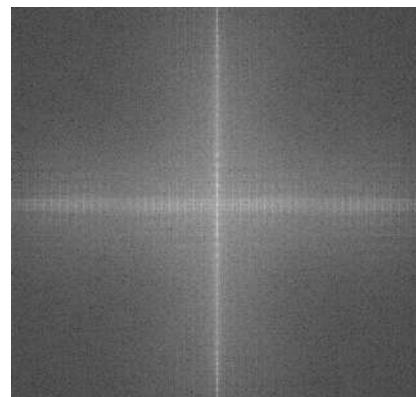
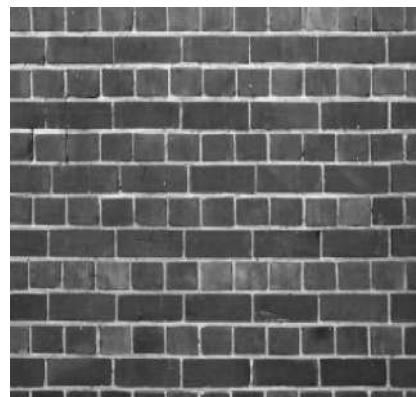


247



247

Modèle fréquentiel : transformée de Fourier

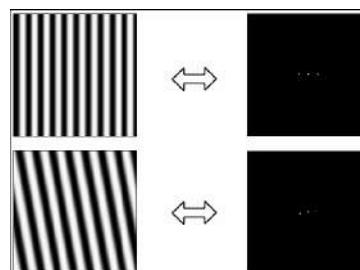
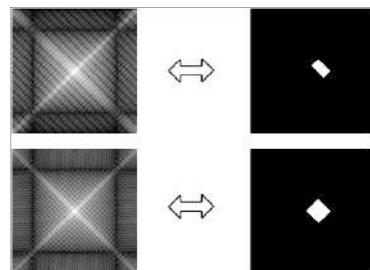
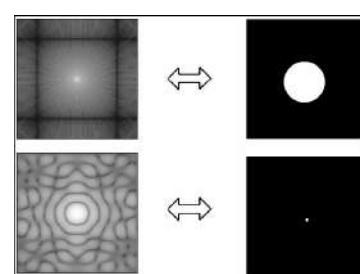
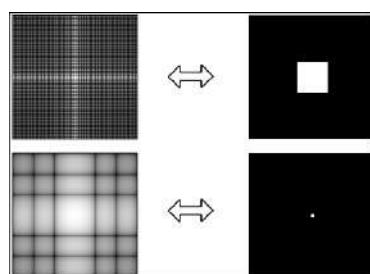


248



248

Modèle fréquentiel : transformée de Fourier



249



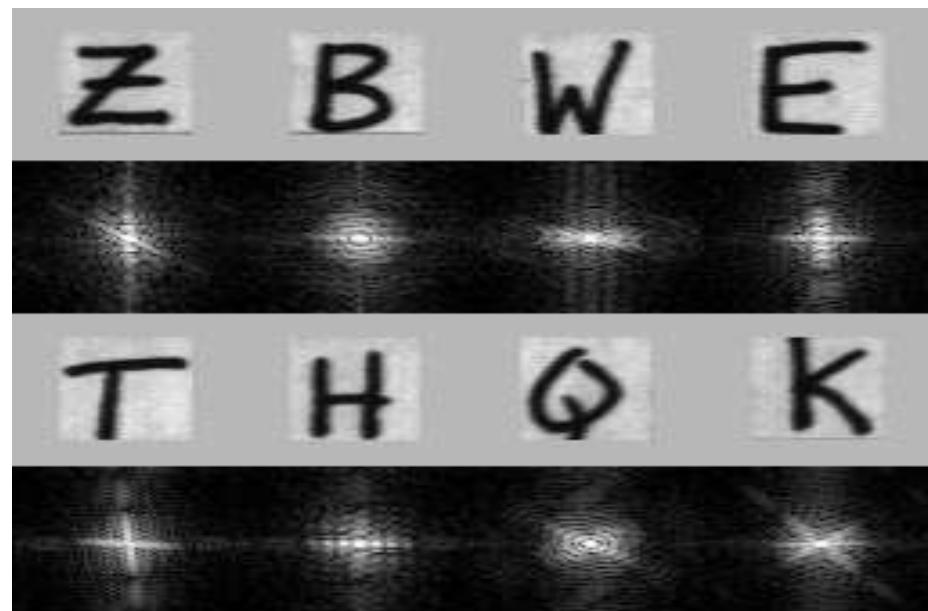
249

Modèle fréquentiel : transformée de Fourier

Reconnaissance
de caractères

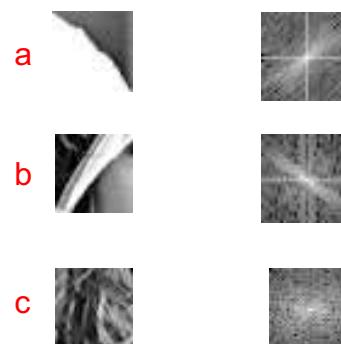
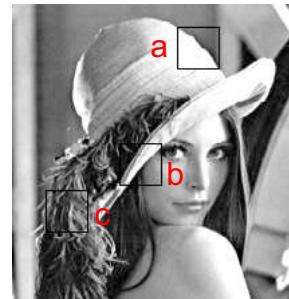
Characters
recognition

250



250

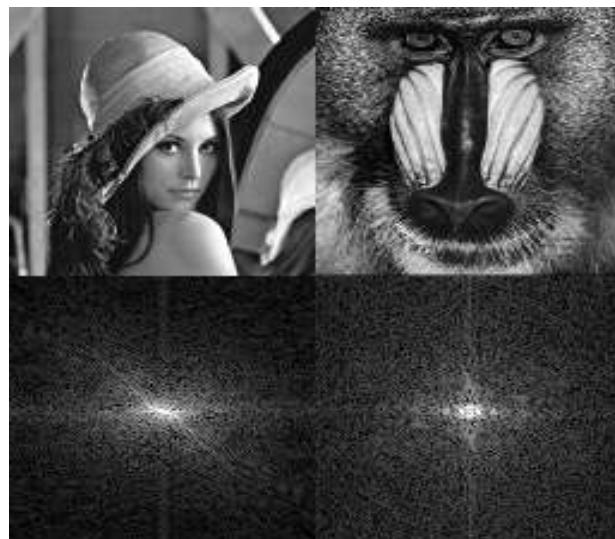
Modèle fréquentiel : transformée de Fourier



251

251

Modèle fréquentiel : transformée de Fourier



252



252

Quizz : transformée de Fourier

Que se passe-t-il sur la transformée de Fourier :

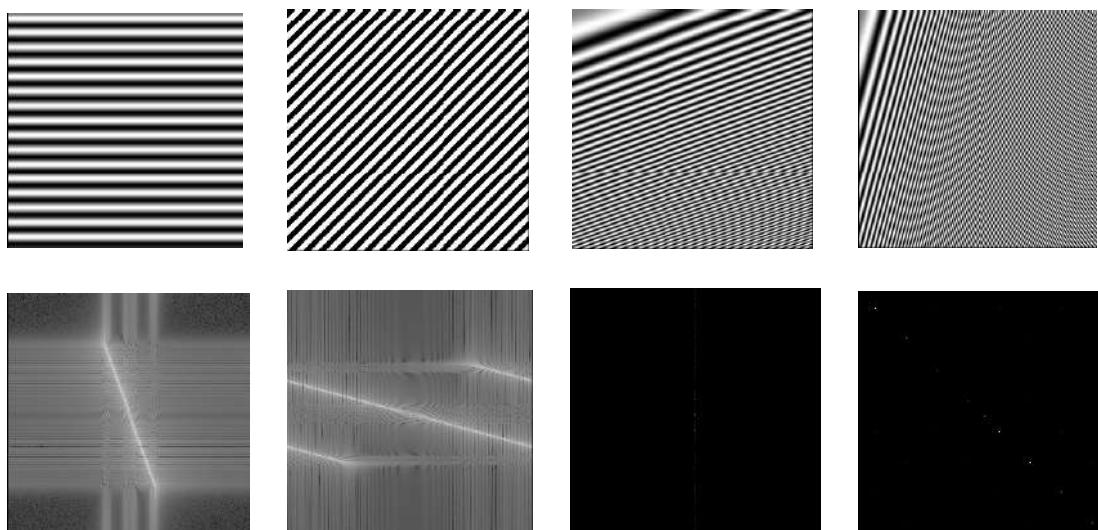
- d'une image qu'on pivote (rotate) ?
idéalement, rotation de la magnitude et de la phase
→ en pratique, effets de bords.
- d'une image qui subit une translation (translate) ?
idéalement, seule la phase change.
- d'une image qu'on réduit proportionnellement (scale) de moitié ?
magnitude découpée (cropée).

253



253

Quizz : associez à chaque image son spectre

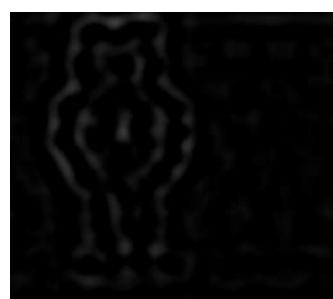


254



254

Filtres fréquentiels



Filtres fréquentiels :

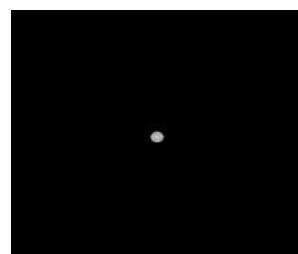
La transformée de Fourier permet essentiellement de faire du filtrage fréquentiel.
→ on annule l'amplitude de certaines fréquences

255



255

Filtre passe bas

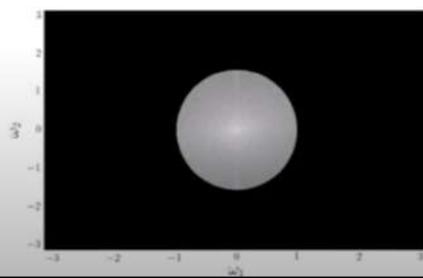
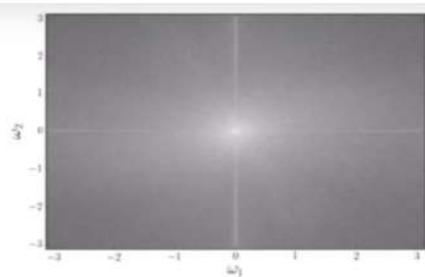


256



256

Filtre passe bas

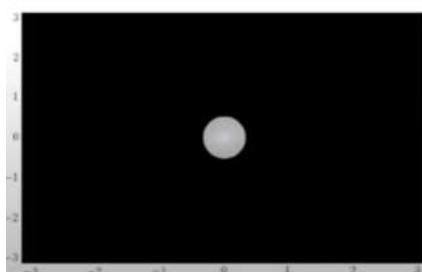
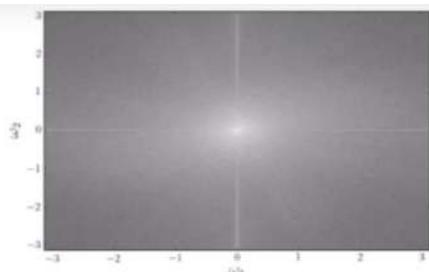


257



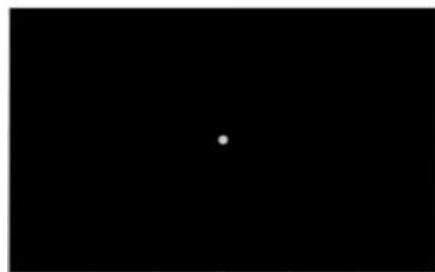
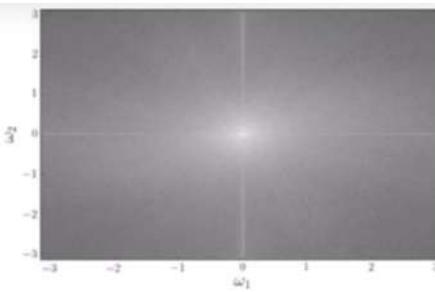
257

Filtre passe bas



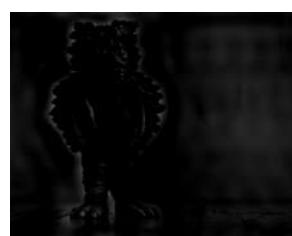
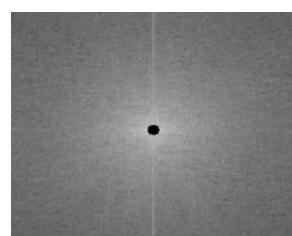
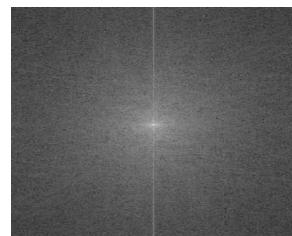
258

Filtre passe bas



259

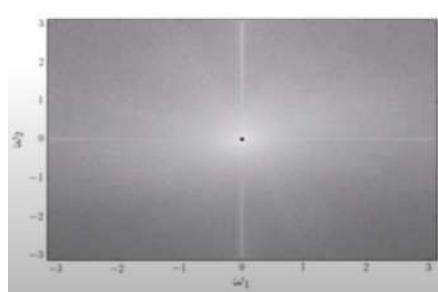
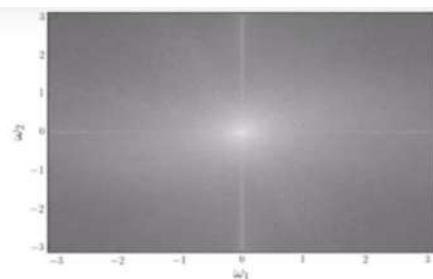
Filtre passe haut



260

260

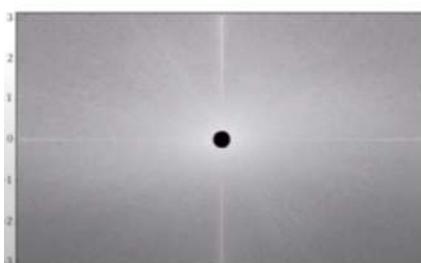
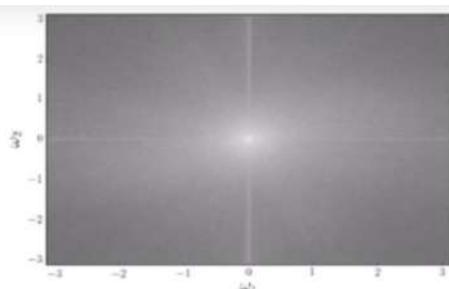
Filtre passe haut



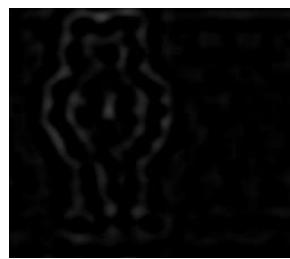
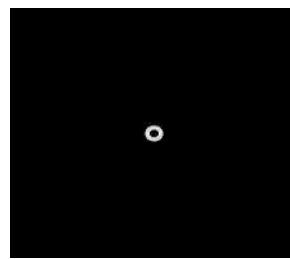
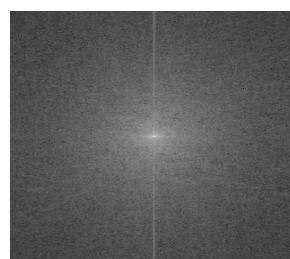
261

261

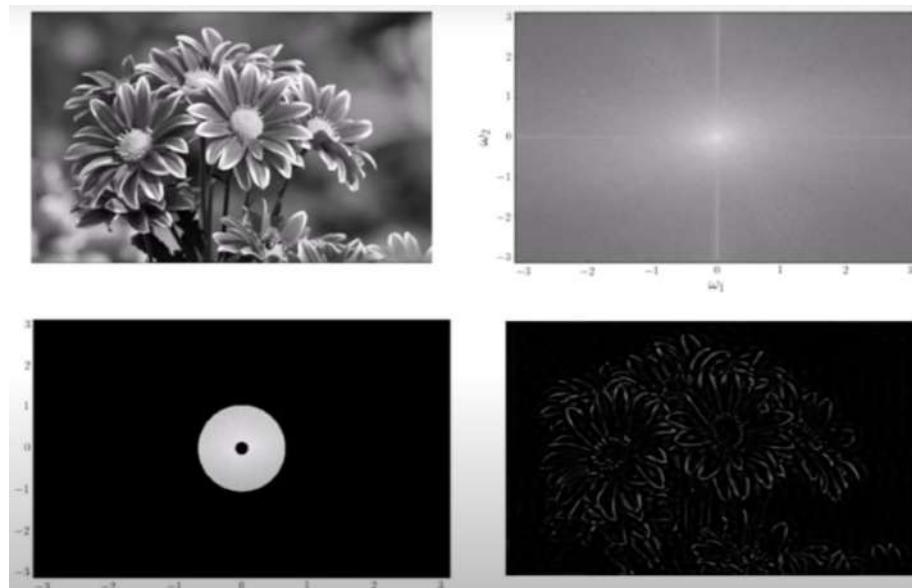
Filtre passe haut



Filtre passe bande

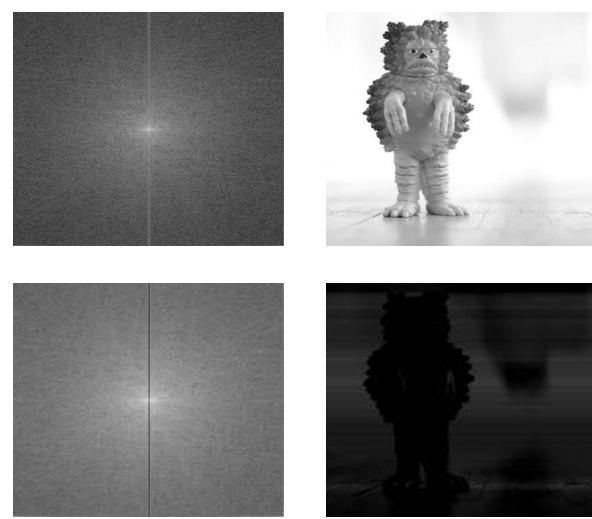


Filtre passe bande



264

Filtre vertical



265

Convolutions et TF

Théorème de convolution :

la convolution de 2 fonctions dans le domaine spatial est équivalente à la multiplication terme à terme dans le domaine fréquentiel.

Plus précisément :

soient $f(x, y)$ une image et $h(x, y)$ un filtre de convolution;
nous avons :

$$\mathbf{F}(f(x, y) * h(x, y)) = \mathbf{F}(f(x, y))\mathbf{F}(h(x, y)) = F(u, v)H(u, v)$$

266



266

Convolutions et TF

En pratique :

soient $f(x, y)$ une image et $h(x, y)$ un filtre de convolution, alors :

- ① $F = \text{DFT}(f)$
- ② redimensionner h à la résolution de f
- ③ $H = \text{DFT}(h)$
- ④ $F' = F \otimes H$ \otimes : multiplication magnitude terme à terme
- ⑤ $f' = \text{IDFT}(F')$

DFT = Discrete Fourier Transform

IDF = Inverse Discrete Fourier Transform

267



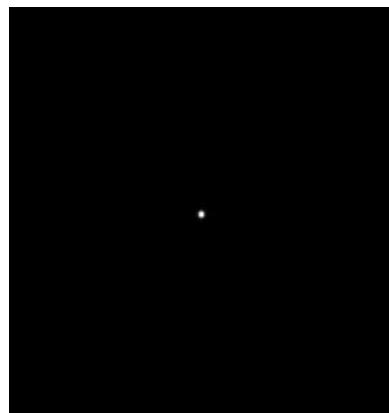
267

126

Filtre gaussien



image originale



kernel gaussien

268



268

Filtre gaussien

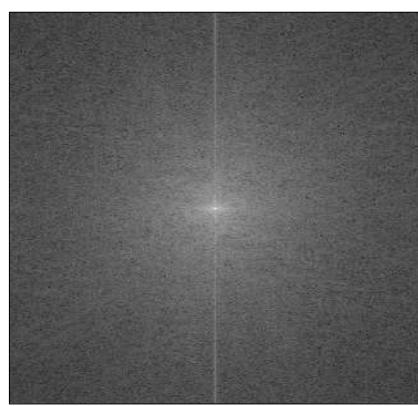
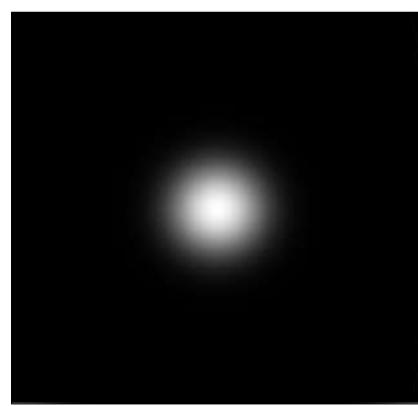


image Fourier



kernel Fourier

269



269

127

Filtre gaussien

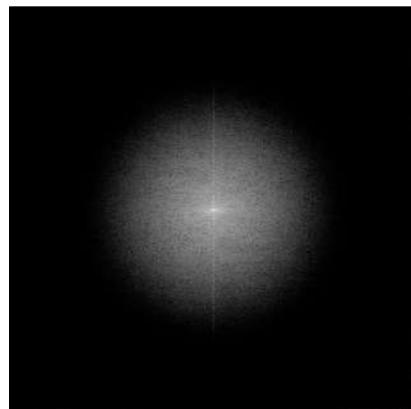
magnitude \otimes kernel

image filtrée

270



270

Filtre Sobel x

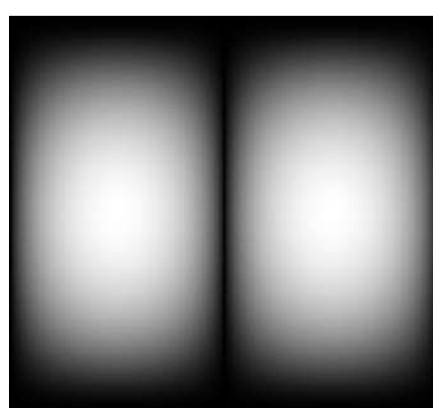
 F_x (kernel)

image filtrée

271

271



128

Filtre Sobel y

$$S_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

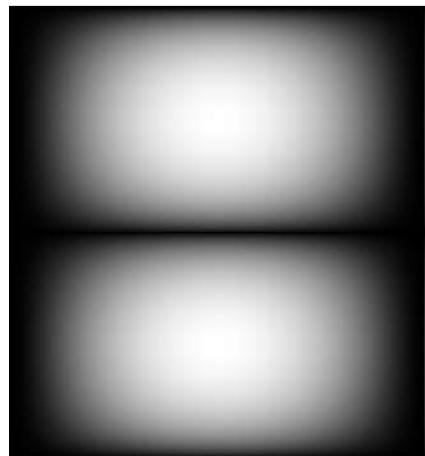
 $F(\text{kernel})$ 

image filtrée

Filtre laplacien

$$\mathcal{L} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

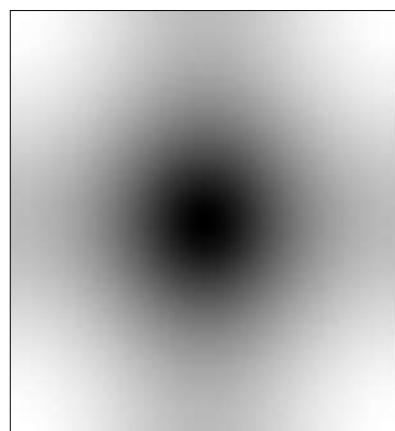
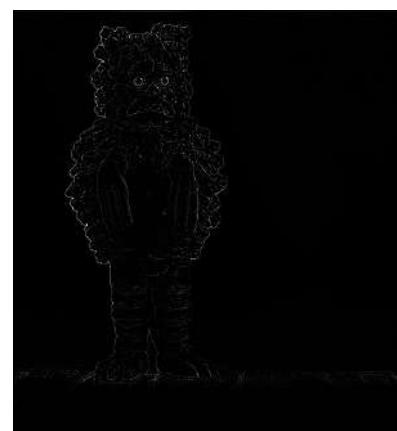
 $F(\text{kernel})$ 

image filtrée

Implémentation

Fast Fourier Transform (FFT) :

- transformation séparable : (plus rapide)
transformation $2D (m \times n) \leftrightarrow m$ transformation $1D (n)$
- pour les images dont la taille est une puissance de 2 :
approche récursive
- complexité classique : $O(n^2) \leftrightarrow$ FFT : $O(n \log n)$
pour un grand nombre de pixels n : grosse différence
- implémentation GPU

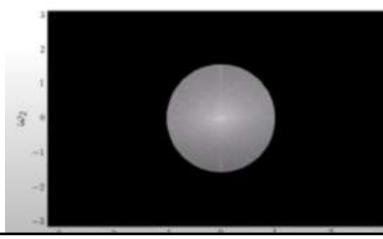
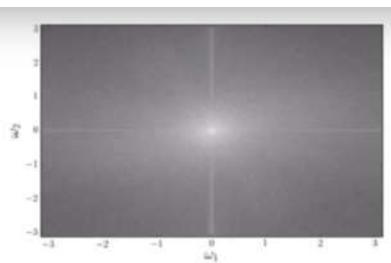
274



274

Fourier Transform

<https://github.com/ssingal05/ImageTransformer>
<https://www.youtube.com/watch?v=oACegp4iGi0>



275



275

130

En pratique



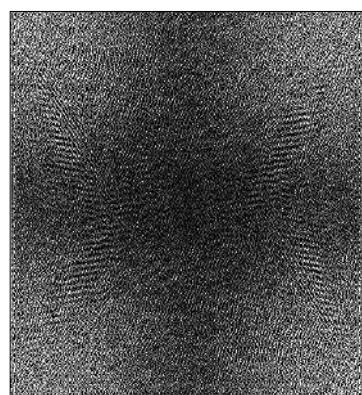
image de départ

276

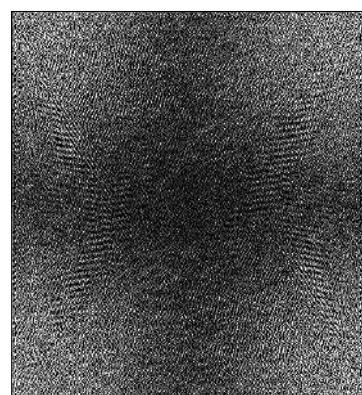


276

DFT



partie réelle



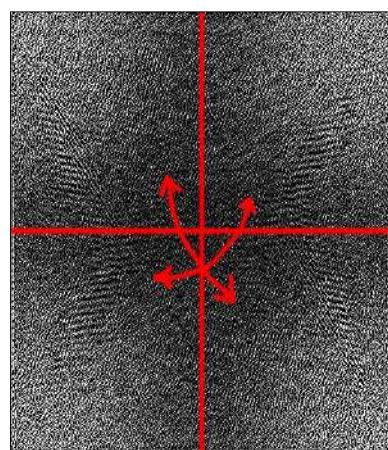
partie imaginaire

277

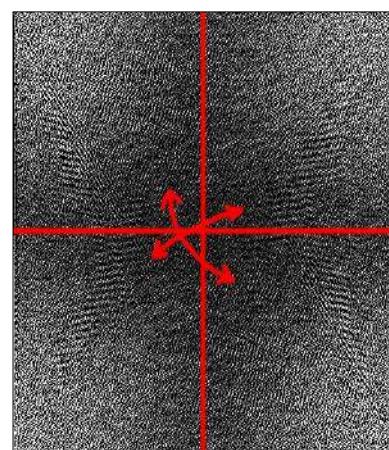


277

Recentrage



partie réelle



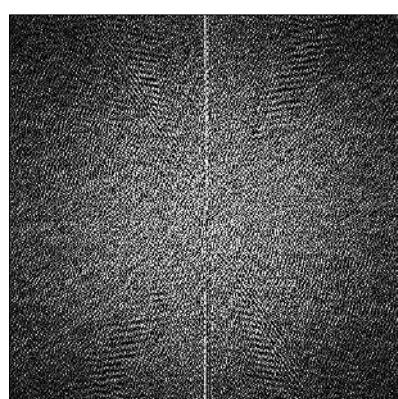
partie imaginaire

278

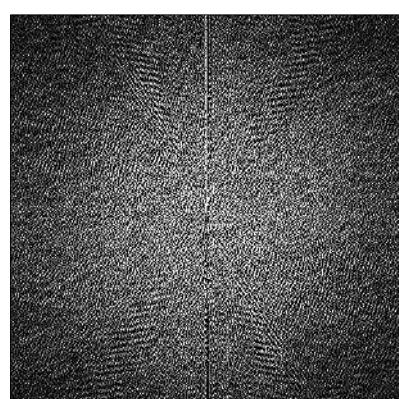


278

Recentrage



partie réelle



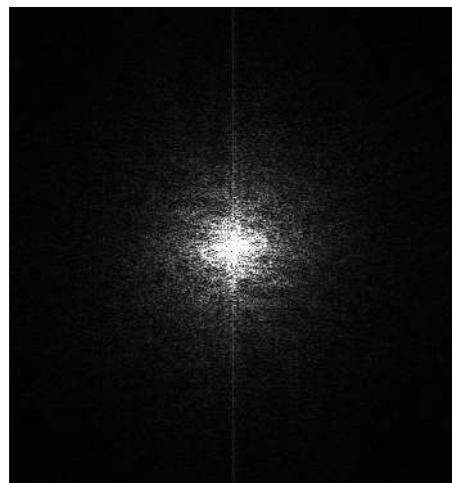
partie imaginaire

279

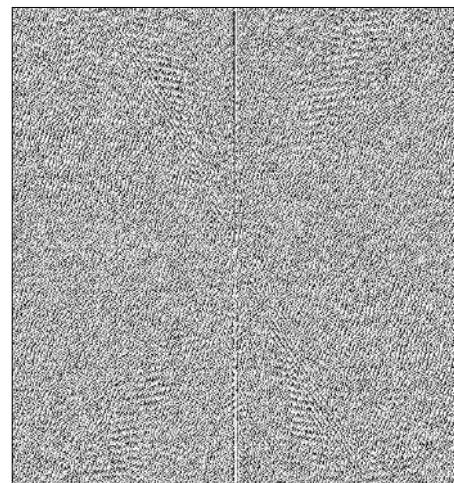


279

Cartésien → polaire



280 amplitude $\sqrt{Re^2 + Im^2}$

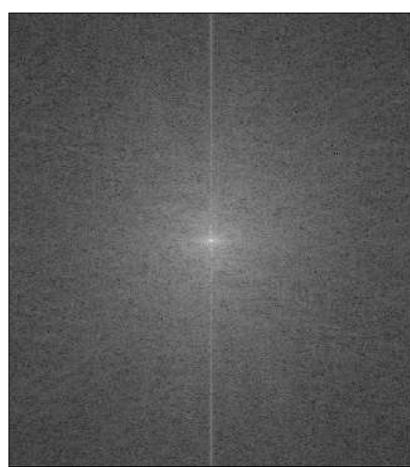


phase : $\varphi = \text{atan}(Im/Re)$

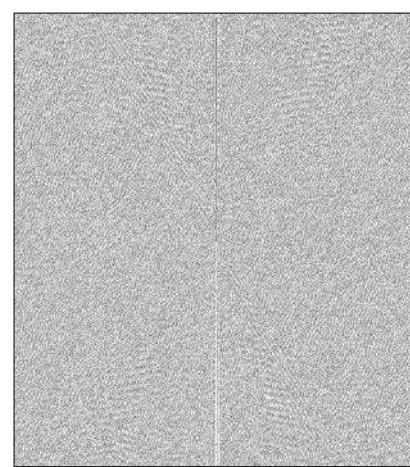


280

Affichage



281 $255 \log(x + 1)/\max$



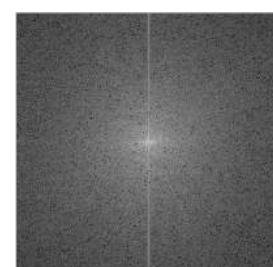
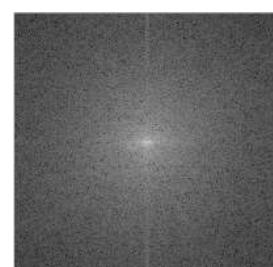
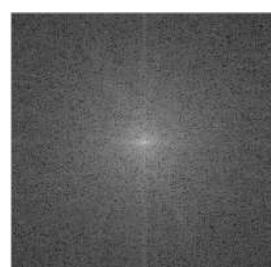
$255x/(2\max + 1) + 127$



281

Et la couleur

RGB



282



282

d. Modèles différentiels

Traitement d'images



283

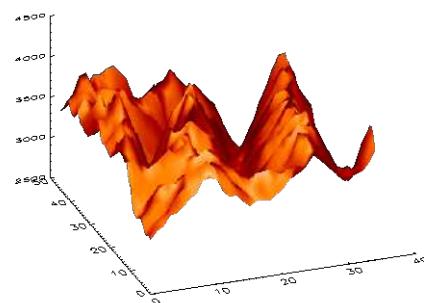
134

Modèle différentiel

Dans le modèle différentiel, on considère l'image comme une fonction continue $f(x,y)$, dont on étudie le comportement local à l'aide de ses dérivées.

Une telle étude, fondée sur la formule de Taylor, n'a de sens que si la fonction f a une certaine régularité, ce qui constitue le problème clef des méthodes différentielles

284

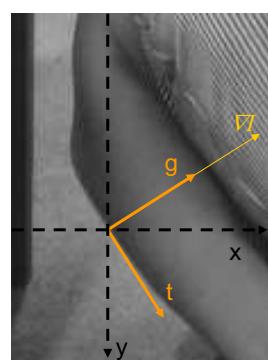


284

Modèle différentiel : gradient de l'image

Au premier ordre, on peut ainsi associer à chaque point (x,y) un repère propre (t,g) , où le vecteur t donne la direction de l'**isophote** (ligne de variation minimale) et g la direction orthogonale, celle du **gradient**.

285



285

Modèle différentiel : gradient de l'image

Grâce au plongement dans le continu, le modèle différentiel permet en outre d'exprimer un grand nombre d'opérations d'analyse en termes **d'équations aux dérivées partielles** (EDP), ce qui permet de donner un fondement mathématique satisfaisant aux traitements et aussi de fournir des méthodes pour les calculer, par des schémas numériques de résolutions :

- Diffusion anisotrope
- Contours actifs
- Ensembles de niveau
-

286



286

Contour actif



287



287

Contour actif



288

https://www.youtube.com/watch?v=CeU_yZjdVqY



288

Contour actif (active contour)

Un **modèle de contour actif**, souvent nommé *snake* (« serpent » en anglais) dans la littérature, est une structure dynamique utilisée en traitement d'image et en vision artificielle.

Ils ont été introduits de manière formelle par Kass et Witkin en 1987.

Plusieurs approches sont possibles et permettent de résoudre le problème de la segmentation et de la détection de contour en utilisant un modèle de courbe déformable qui épouse la forme des objets.

289



289

Contour actif (active contour)

Snakes (serpents) – modèles actifs déformables



290



Exemple d'un snake qui épouse le corps de l'araignée



290

Contour actif (active contour)

$$E_{\text{snake}}^* = \int_0^1 E_{\text{snake}}(\mathbf{v}(s)) ds = \int_0^1 (E_{\text{internal}}(\mathbf{v}(s)) + E_{\text{image}}(\mathbf{v}(s)) + E_{\text{con}}(\mathbf{v}(s))) ds$$

Energie interne (élastique) - contrôler les déformations du serpent
 Energie externe (bords) - contrôlent l'ajustement du contour à l'image
 Energie contrainte – introduite par l'utilisateur



292



292

Contour actif (active contour)

$$E_{\text{snake}}^* = \int_0^1 E_{\text{snake}}(\mathbf{v}(s)) ds = \int_0^1 (E_{\text{internal}}(\mathbf{v}(s)) + E_{\text{image}}(\mathbf{v}(s)) + E_{\text{con}}(\mathbf{v}(s))) ds$$

E_{internal} = Energie interne (élastique)

$E_{\text{internal}} = E_{\text{cont}} + E_{\text{curv}}$ (continuité et lissage)

$$E_{\text{internal}} = \frac{1}{2}(\alpha(s)|\mathbf{v}_s(s)|^2) + \frac{1}{2}(\beta(s)|\mathbf{v}_{ss}(s)|^2) = \frac{1}{2}\left(\alpha(s)\left\|\frac{d\bar{v}}{ds}(s)\right\|^2 + \beta(s)\left\|\frac{d^2\bar{v}}{ds^2}(s)\right\|^2\right)$$

E_{image} = Energie de l'image (bords) - lignes, bords, et terminaisons présentes dans l'image :

$$E_{\text{image}} = w_{\text{line}} E_{\text{line}} + w_{\text{edge}} E_{\text{edge}} + w_{\text{term}} E_{\text{term}},$$

E_{cont} = Energie contrainte (utilisateur)

293

Contour actif (active contour)

$$E_{\text{snake}}^* = \int_0^1 E_{\text{snake}}(\mathbf{v}(s)) ds = \int_0^1 (E_{\text{internal}}(\mathbf{v}(s)) + E_{\text{image}}(\mathbf{v}(s)) + E_{\text{con}}(\mathbf{v}(s))) ds$$

Optimisation (minimisation) par **descente de gradient**

$$\bar{v}_i \leftarrow \bar{v}_i + F_{\text{snake}}(\bar{v}_i)$$

F_{snake} est la force sur le serpent, définie par l'opposé du gradient du champ d'énergie :

$$F_{\text{snake}}(\bar{v}_i) = -\nabla E_{\text{snake}}(\bar{v}_i) = -\left(w_{\text{internal}} \nabla E_{\text{internal}}(\bar{v}_i) + w_{\text{external}} \nabla E_{\text{external}}(\bar{v}_i)\right)$$

En supposant que les poids $\alpha(s)$ et $\beta(s)$ sont constants par rapport à s , cette méthode itérative peut être simplifiée comme suit :

$$\bar{v}_i \leftarrow \bar{v}_i - \gamma \left\{ w_{\text{internal}} \left[\alpha \frac{\partial^2 \bar{v}}{\partial s^2}(\bar{v}_i) + \beta \frac{\partial^4 \bar{v}}{\partial s^4}(\bar{v}_i) \right] + \nabla E_{\text{ext}}(\bar{v}_i) \right\}$$

295

Contour actif (active contour)

$$E_{\text{snake}}^* = \int_0^1 E_{\text{snake}}(\mathbf{v}(s)) ds = \int_0^1 (E_{\text{internal}}(\mathbf{v}(s)) + E_{\text{image}}(\mathbf{v}(s)) + E_{\text{con}}(\mathbf{v}(s))) ds$$

En pratique : approximation discrète

En pratique, les images ont une resolution finie et peuvent être intégrées seulement par pas temporels finis \mathcal{T} . Par consequent, une approximation discrète doit être faite pour une implementation pratique du serpent.

$$E_{\text{snake}}^* \approx \sum_1^n E_{\text{snake}}(\bar{v}_i) \quad \text{La fonction d'énergie du serpent peut être approximée en utilisant les points discrets du serpent.}$$

$$\text{Par conséquent, les forces du serpent peuvent être approximées comme suit :} \quad F_{\text{snake}}^* \approx - \sum_{i=1}^n \nabla E_{\text{snake}}(\bar{v}_i).$$

297



297

Contour actif

Implémentation

Pseudocode implémentant la méthode snake (serpent) dans sa forme générale

```

function v = snakes (I, v)
    % INPUT: N by M image I, a contour v of n control points
    % OUTPUT: converged contour v of n control points

    E_image = generateImageEnergy (I);

    while not converged
        F_cont = weight.alpha * contourDerivative(v, 2);
        F_curv = weight.beta * contourDerivative(v, 4);
        F_image = interp2 (E_image, v(:,2), v(:,1));
        F_image_norm = weight.k * F_image ./ norm (F_image);
        F_con = inputForces();

        F_internal = F_cont + weight.external * F_curv;
        F_external = weight.external * (F_image + F_con);

        v = updateSnake(v, F_internal, F_external);

        checkConvergence ();
    end

```

299



299

140

où `generateImageEnergy (I)` peut s'écrire sous la forme

```
function E_image = generateImageEnergy (I)
    [C, Cx, Cy, Cxx, Cxy, Cyy] = generateGradients (I);

    E_line = I;
    E_edge = -(Cx.^2 + Cy.^2)^0.5;
    E_term = (Cyy.*Cx.^2 - 2*Cxy.*Cx.*Cy + Cxx.*Cy.^2)./((1 + Cx.^2 + Cy.^2).^(1.5));

    E_image = weight.line * E_line + weight.edge * E_edge + weight.term * E_term;
end
```

300

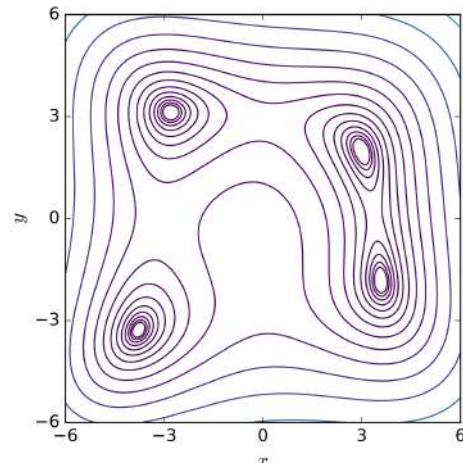
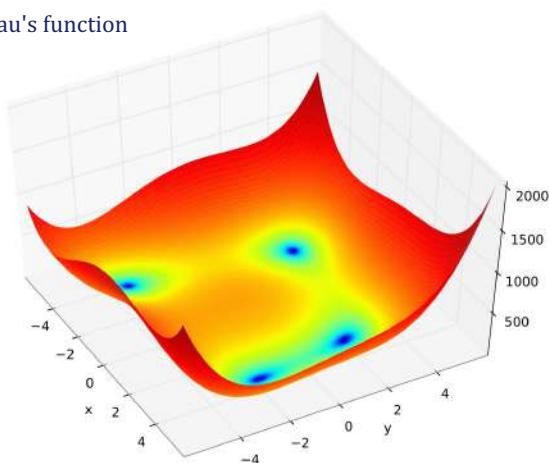


300

Ensembles de niveau (level sets)

Log-spaced level curve plot

Himmelblau's function



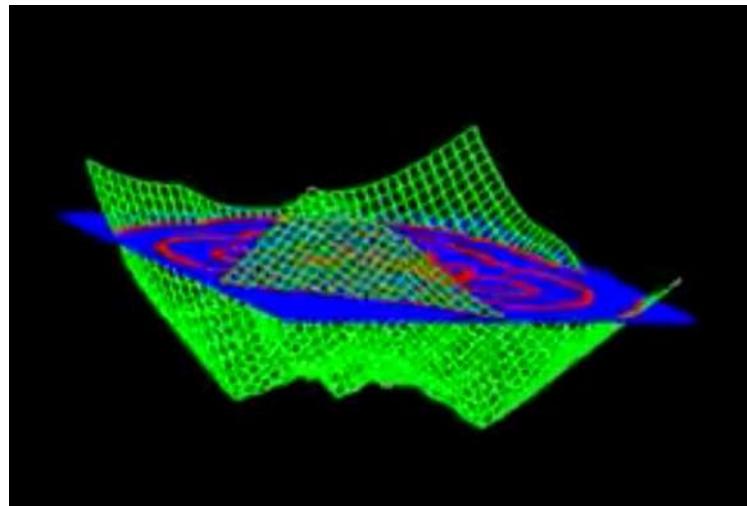
301

301



Ensembles de niveau (level sets)

$$L_c(f) = \{(x_1, \dots, x_n) \mid f(x_1, \dots, x_n) = c\},$$



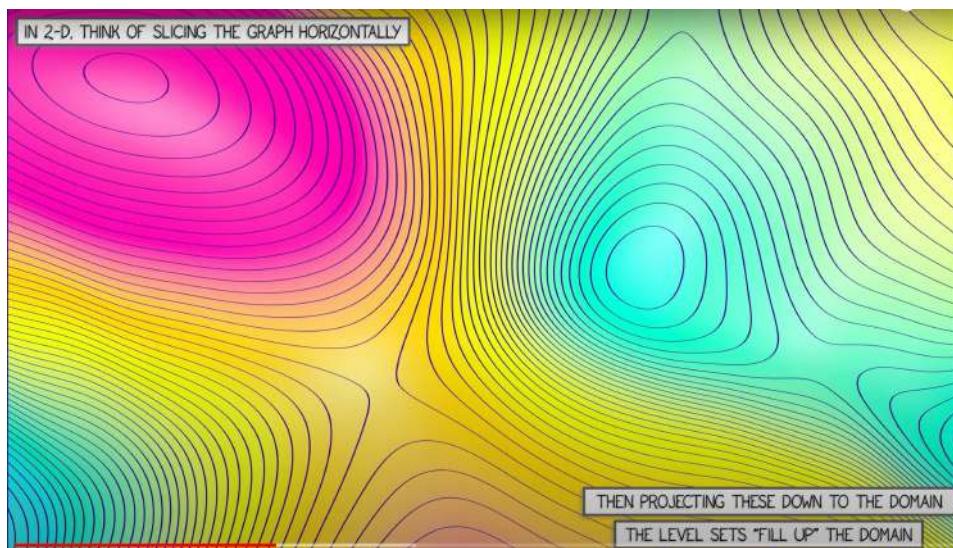
302

<https://www.youtube.com/watch?v=YhqhuhfEPWQ>



302

Ensembles de niveau (level sets)



303

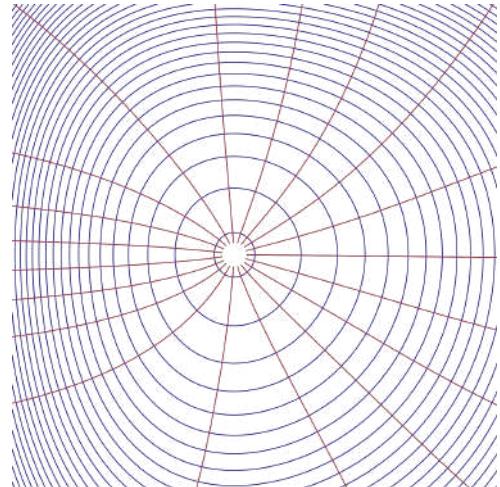
<https://www.youtube.com/watch?v=zEvGUXXW1BI>



303

Ensembles de niveau (level sets)

Considérons une fonction f dont le graphe ressemble à une montagne. Les courbes bleues représentent alors les courbes de niveau. Les courbes rouges suivent la direction du gradient.



304

304

Ensembles de niveau (level sets)

- Géométries implicites, évolution de l'interface par résolution d'EDP.
- Inventés en 1988 by Osher et Sethian:
 - Stanley Osher and James A. Sethian. Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations. *J. Comput. Phys.*, 79(1):12–49, 1988.
- Ouvrages de référence :
 - James A. Sethian. *Level set methods and fast marching methods*. Cambridge University Press, Cambridge, second edition, 1999.
 - Stanley Osher and Ronald Fedkiw. *Level set methods and dynamic implicit surfaces*. Springer-Verlag, New York, 2003.

306

306



Ensembles de niveau

Intuition

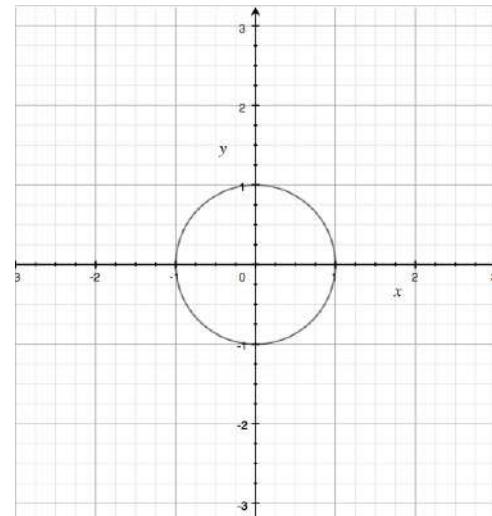
Supposons que nous jetions une pierre au milieu d'un étang. Que se passe-t-il ?

Il y aura une vague d'eau (pour simplifier, choisissons une seule vague), partant de l'épicentre et s'étendant jusqu'à ce qu'elle se dissipe ou atteigne le bord de l'étang.

Comment modéliser et simuler ce phénomène ?

Nous pourrions faire ceci : modéliser la courbe de cette ondulation et suivre son mouvement. Disons qu'au moment $t=1$, voici à quoi ressemble l'ondulation :

308



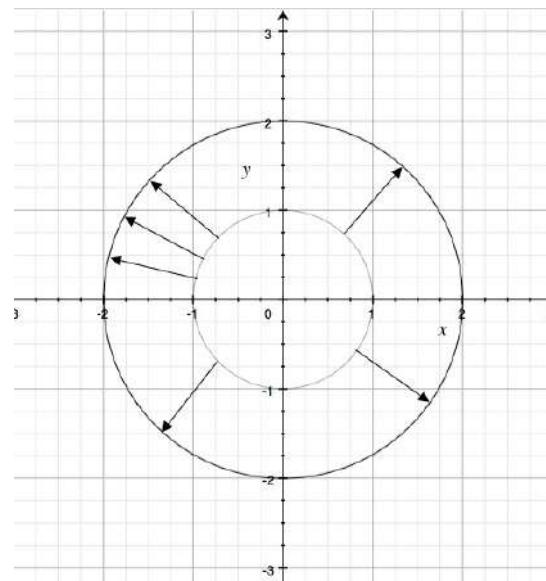
308

Ensembles de niveau

Maintenant, nous voulons modéliser le mouvement au fil du temps. Pour ce faire, nous devons suivre le mouvement de la courbe ci-dessus.

Une façon de le faire est d'échantillonner suffisamment de points dans cette courbe et de la déplacer dans la direction normale à la courbe.

310



310

144

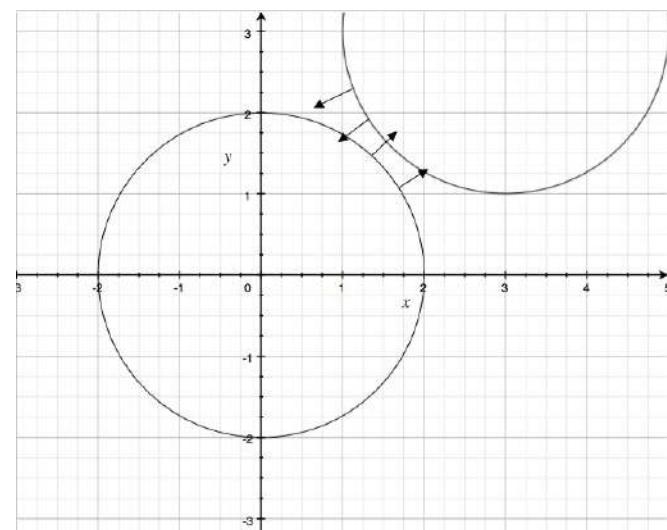
Ensembles de niveau

C'est une bonne solution pour une simulation très simple (comme celle-ci). Cependant, considérez ce qui se passe dans ce cas :

En supposant qu'il n'y a pas de force externe, ces deux courbes vont fusionner en une seule courbe. Nous devons également envisager le cas où une courbe se divise en deux ou plusieurs courbes. Comment modéliser cela ?

C'est là que le LSM fait merveille. Au lieu de modéliser la courbe de manière **explicite**, les EdN la modélise de manière **implicite**. Mais comment peut-il nous aider à modéliser la division, la fusion et le mouvement de la courbe ? Voyons comment cela fonctionne.

312

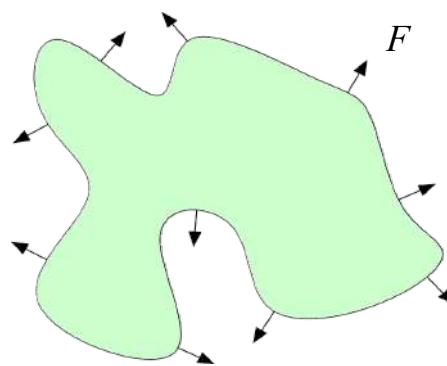


S UNIVERSITÉ

312

Courbes et surfaces évolutives

- Propager la courbe selon la fonction de vitesse $v = F \mathbf{n}$
- F dépend de l'espace, du temps, et de la courbe elle-même
- Surfaces en trois dimensions



S SORBONNE UNIVERSITÉ

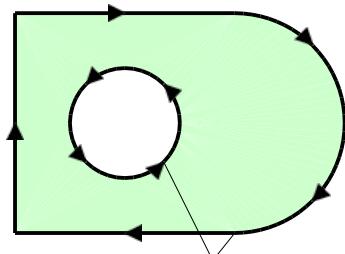
314

145

Geometry Representations

Explicit Geometry

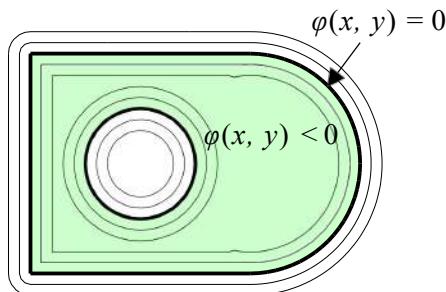
- Parameterized boundaries



$$(x, y) = (x(s), y(s))$$

Implicit Geometry

- Boundaries given by zero level set



$$\varphi(x, y) > 0$$

316



316

Techniques Explicites

- Approche simple : représenter explicitement les courbes par des nœuds $x^{(i)}$ et des lignes
- Propager la courbe par résolution de EDP

$$\frac{d\mathbf{x}^{(i)}}{dt} = \mathbf{v}(\mathbf{x}^{(i)}, t), \quad \mathbf{x}^{(i)}(0) = \mathbf{x}_0^{(i)},$$

- Vecteur normal, courbure, etc par approximations différentielles, e.g.:

$$\frac{d\mathbf{x}^{(i)}}{ds} \approx \frac{\mathbf{x}^{(i+1)} - \mathbf{x}^{(i-1)}}{2\Delta s}$$

317

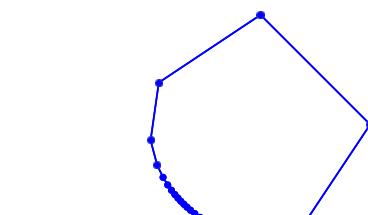


317

Techniques explicites - Inconvénients

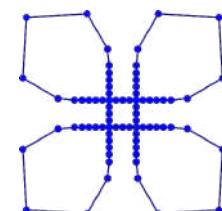
- La redistribution des noeuds est nécessaire, ce qui introduit des erreurs
- Pas de solution entropique, les angles aigus ne sont pas traités correctement
- Nécessité d'un traitement spécial pour les changements de topologie
- Contraintes de stabilité pour les fonctions de vitesse dépendant de la courbure

Distribution des noeuds

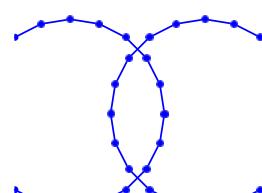


319

Angles vifs



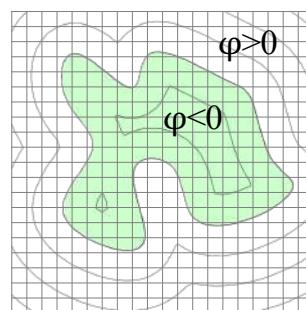
Modifications de la topologie



319

Implicit Geometries

- Represent curve by zero level set of a function, $\varphi(x) = 0$
- Special case: *Signed distance function*:
 - $|\nabla \varphi| = 1$
 - $|\varphi(x)|$ gives (shortest) distance from x to curve



321

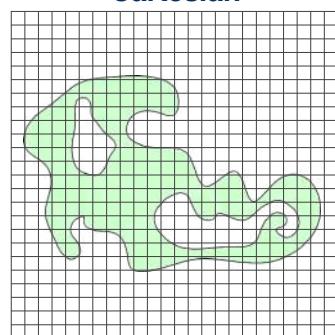
321



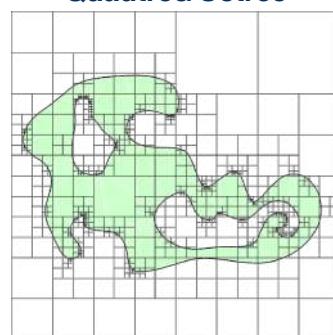
Discretized Implicit Geometries

- Discretize implicit function φ on *background grid*
- Obtain $\varphi(x)$ for general x by interpolation

Cartesian



Quadtree/Octree



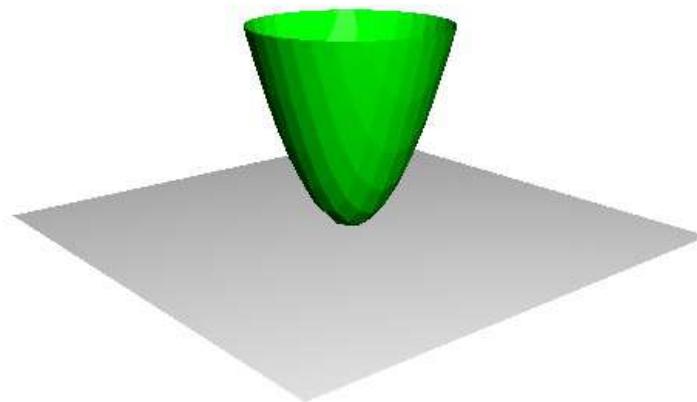
322



322

Ensembles de niveau

Supposons que nous ayons cette courbe 3D (surface) :



323



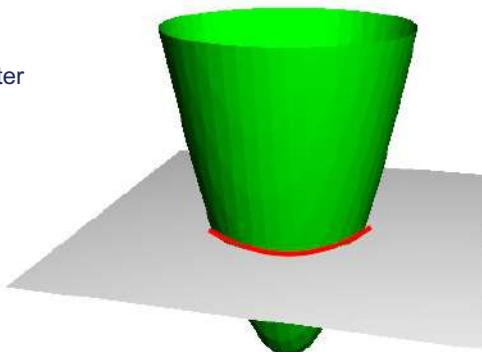
323

Ensembles de niveau

Nous pouvons modéliser la courbe (cercle) ci-dessus avec cette courbe en exploitant la relation entre surface, plan et courbe.

Ce que nous allons faire, c'est ajuster notre surface pour qu'elle croise un plan à une certaine hauteur.

Comme ceci :



325



325

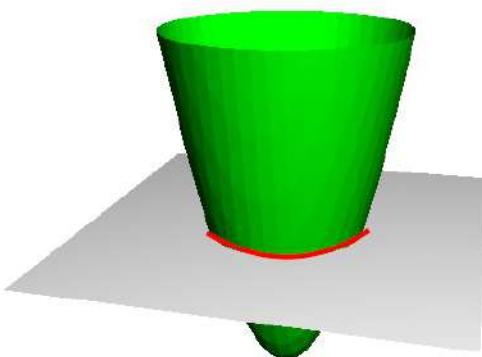
Ensembles de niveau

Regardez de plus près l'intersection.
Qu'est-ce que c'est ?

Ce n'est autre qu'une courbe, plus précisément un cercle !

Cette courbe est la courbe de niveau, c'est-à-dire la courbe d'un ensemble de niveaux.

C'est l'idée qui sous-tend les EdN.
Nous modifions implicitement notre courbe en transformant la surface puis en l'intersectant à un plan et nous évaluons la courbe de niveau qui en résulte.



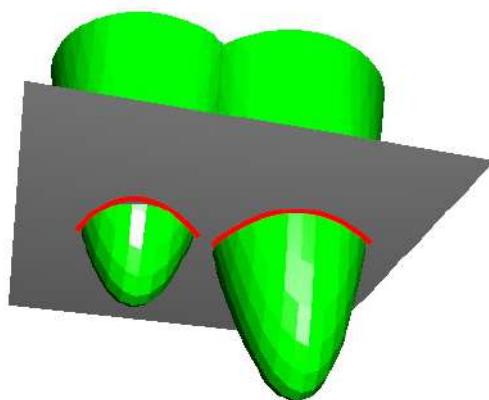
327

327

Ensembles de niveau

How do LSM could model the merge and split operation of curves?

Let's do something to our surface and see what does it do to our level curve.



329



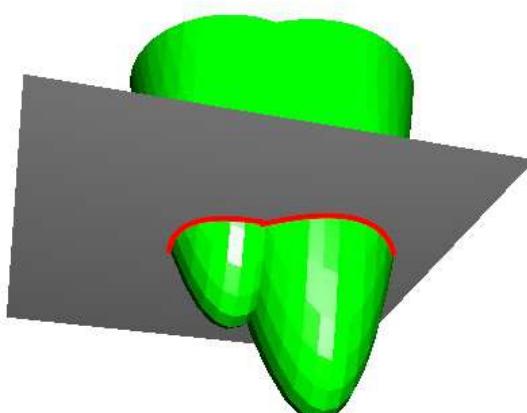
329

Ensembles de niveau

Dans le graphique ci-dessus, nous transformons la surface en une surface avec deux minima.

Nous pouvons voir l'implication au niveau de la courbe de niveau, au lieu d'un seul cercle, maintenant il devient deux.

De même avec l'opération de fusion :



331

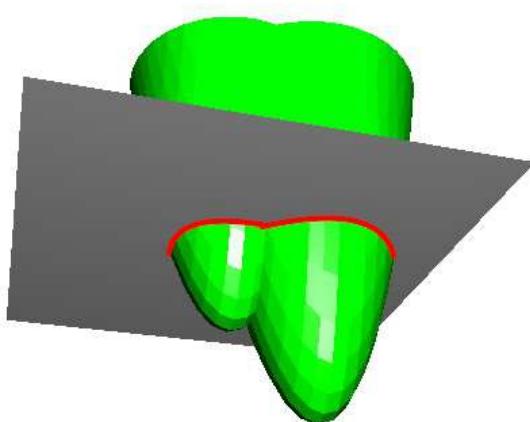


331

Ensembles de niveau

Sans effort, la courbe de niveau
le capte !

C'est une intuition puissante et
nous allons la formuler.



333

333

Ensembles de niveau

Supposons que nous ayons une surface $\phi(x)$.

L'ensemble des niveaux c de cette surface est donné par :

$$\{x \mid \phi(x) = c\}$$

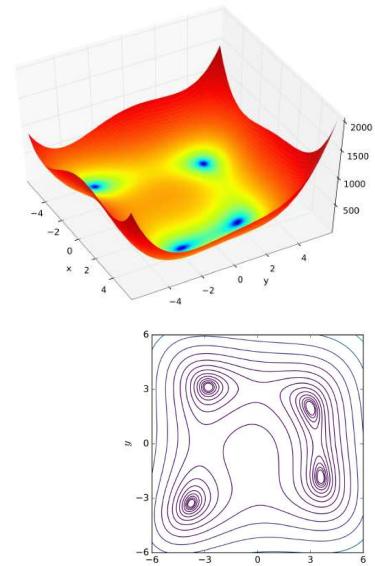
Formellement, nous voulons suivre la courbe de niveau à $c = 0$, qui est l'ensemble de niveau zéro de ϕ .

$$\{x \mid \phi(x)=0\}$$

Comme nous traitons de l'évolution de la courbe et de la surface, nous allons paramétriser notre surface avec une variable temporelle t telle que :

$$\phi(x(t),t)=0$$

335



335

Ensembles de niveau

Nous pourrions penser que c'est la surface à l'instant t , étant donné la variable x à l'instant t .

Ensuite, comme nous voulons suivre le mouvement de la courbe de niveau zéro de ϕ , nous allons le dériver par rapport à t , c'est-à-dire que nous dérivons l'équation du mouvement de ϕ .

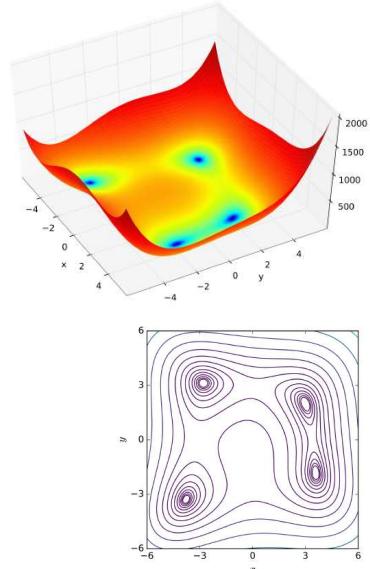
La dérivation de la position est la vitesse, et connaissant la vitesse, nous pourrions modéliser le mouvement de la surface.

$$\frac{\partial \phi(x(t), t)}{\partial t} = 0$$

En utilisant la dérivation des fonctions composées, on obtient :

$$\frac{\partial \phi}{\partial x(t)} \frac{\partial x(t)}{\partial t} + \frac{\partial \phi}{\partial t} = 0$$

337



337

Ensembles de niveau

Rappelez-vous que, par définition, la dérivée partielle la plus à gauche est le gradient de notre surface. Et aussi, pour une lecture plus claire, nous allons changer l'équation de Leibniz en une notation plus compacte :

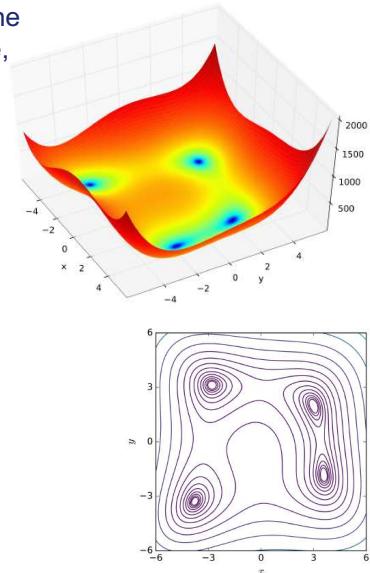
$$\nabla \phi X_t + \phi_t = 0$$

Comme nous l'avons dit plus haut, la direction du mouvement de la courbe est normale, c'est-à-dire $\nabla \phi / \|\nabla \phi\|$. Bien sûr, il y a aussi une force qui déplace la courbe, nous l'appelons F . Par conséquent, le vecteur vitesse est donné par $x_t = F \cdot \nabla \phi / \|\nabla \phi\|$.

$$\nabla \phi F \frac{\nabla \phi}{\|\nabla \phi\|} + \phi_t = 0$$

$$F \|\nabla \phi\| + \phi_t = 0$$

339



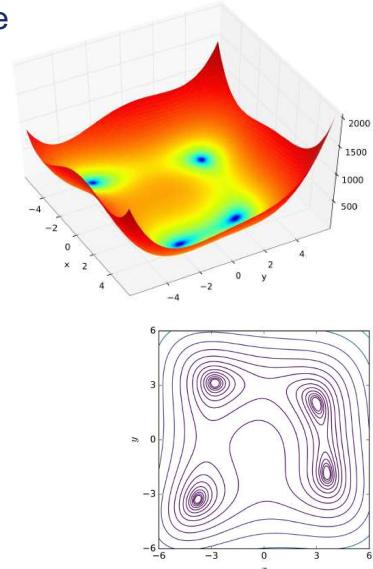
339

Ensembles de niveau

Enfin, pour organiser un peu les choses, nous avons notre équation de niveau :

$$\phi_t = -F\|\nabla\phi\|$$

Cela nous donne la vitesse de l'évolution de la surface de ϕ .



341

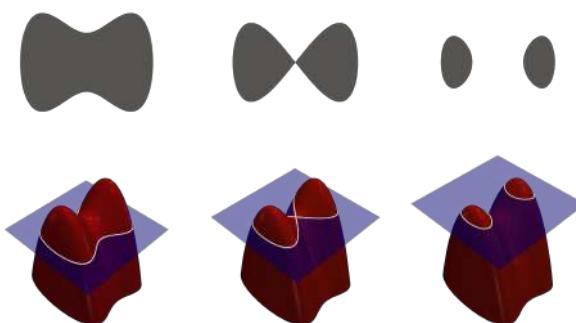
341

Ensembles de niveau

La méthode des ensembles de niveaux permet de suivre très facilement les formes qui changent de **topologie**,

Par exemple, lorsqu'une forme se divise en deux, développe des trous, ou l'inverse de ces opérations.

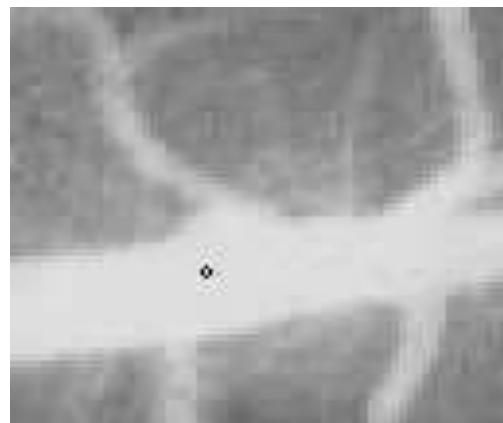
Tout cela fait de la méthode des niveaux et ensembles un outil formidable pour modéliser des objets qui varient dans le temps, comme le gonflage d'un **airbag** ou une goutte d'huile flottant dans l'eau.



343

343

Ajustement avec des ensembles de niveau



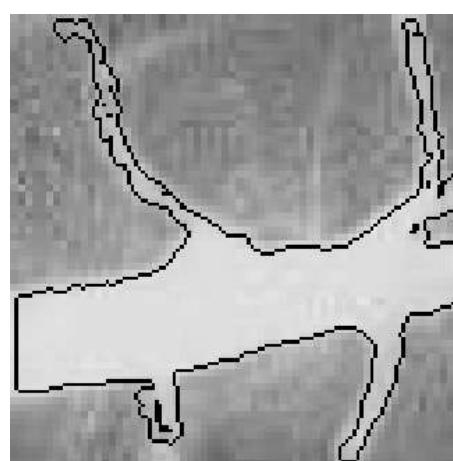
Vein segmentation

345



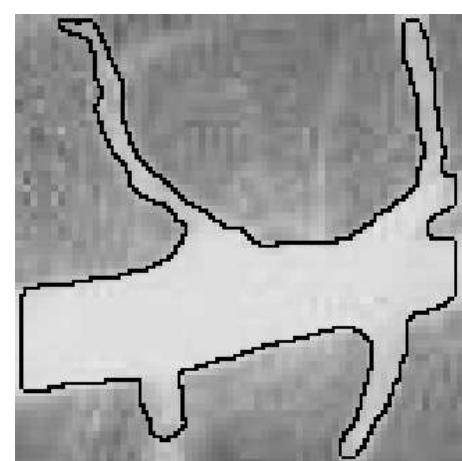
345

Ajustement avec des ensembles de niveau



No level set tuning

346



With level set tuning

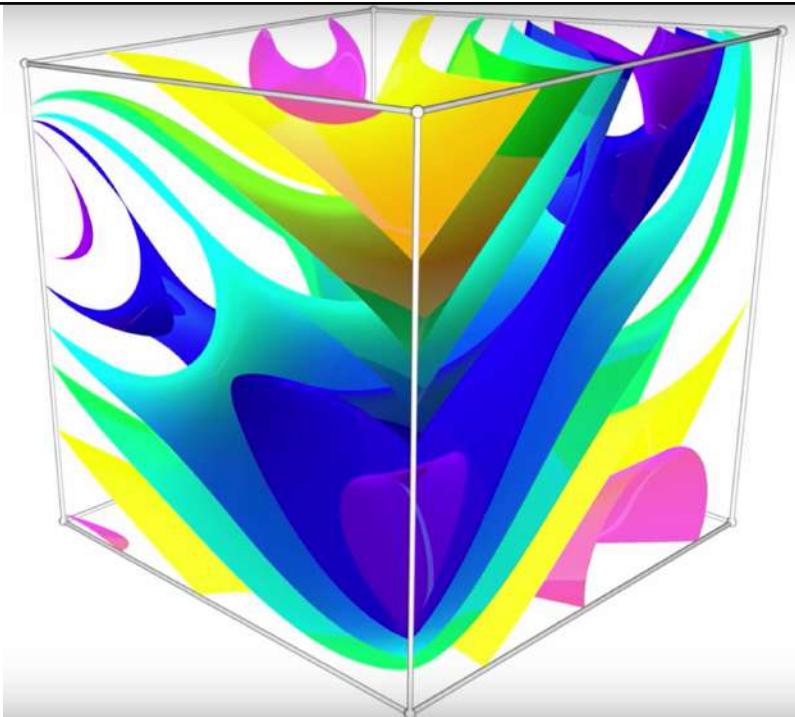


346

3D Level sets

<https://www.youtube.com/watch?v=zEvGUXXW1BI>

347

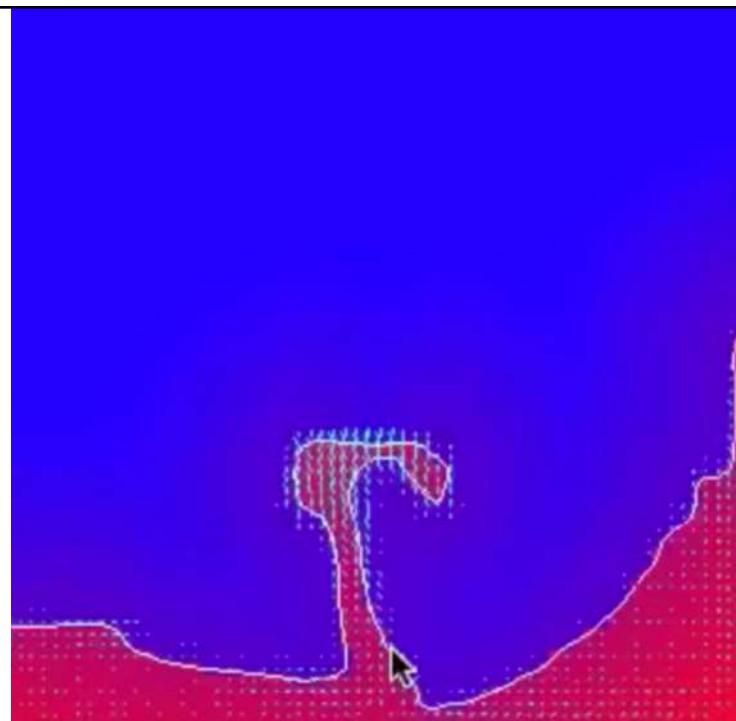


347

2D water simulation using Level Set Method

https://www.youtube.com/watch?v=lye_L9uLS7k

348



348

Topology Optimization

(Level Set Method, Phase Field Method, FEM)

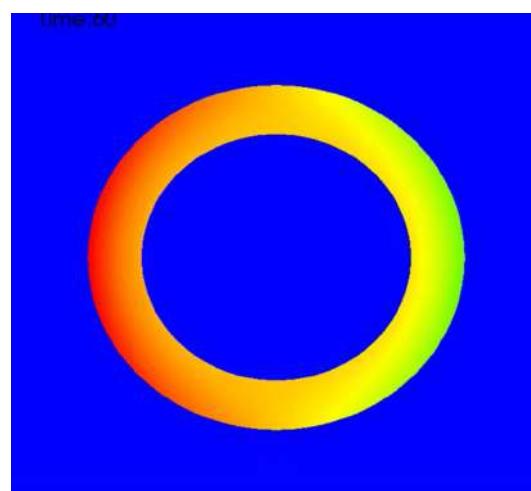
https://www.youtube.com/watch?v=Rl6t_gWb3oU



349

349

Diffusion anisotrope de la chaleur (anisotropic diffusion of heat)



350

https://www.youtube.com/watch?v=VwVRprmq_8w



350

e. Modèles ensemblistes

Set theory models



Traitement d'images

351

Modèle ensembliste

En **morphologie mathématique**, l'image est considérée comme un **ensemble**, dont on étudie les propriétés en fonction de relations locales avec un ensemble de référence (élément structurant) en termes d'*intersection* et d'*inclusion* (relations en tout-ou-rien).

Les opérateurs morphologiques élémentaires sont :

L'érosion

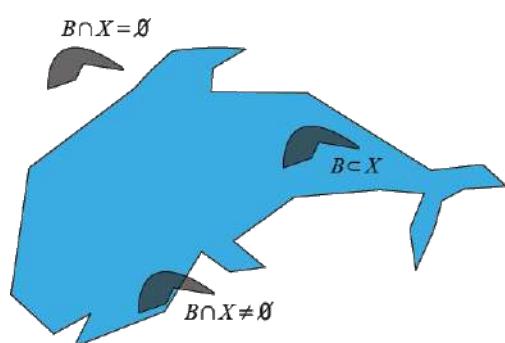
La dilatation

D'autres opérateurs plus complexes peuvent être obtenus en combinant l'érosion et la dilatation :

L'ouverture

La fermeture

La squelettisation fait aussi partie des opérateurs morphologiques



352

352

Modèle ensembliste

La morphologie mathématique binaire

- ◆ Modification d'un ensemble géométrique
- ◆ Application d'un élément de morphologie de géométrie connue appelé **élément structurant**, centré en chaque pixel
- ◆ Opérateur de la théorie des ensembles : union, intersection, inclusion, exclusion, complémentation de l'élément...
- ◆ Utilisation des opérateurs min et max pour les traitement des images :
 - Convention : les pixels en blanc (état logique 1) représentent les formes et les pixels en noir (état logique 0) représentent le fond.

353



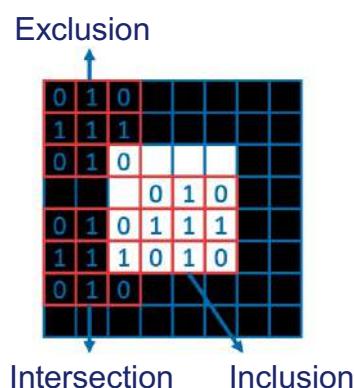
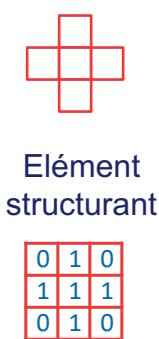
353

Modèle ensembliste

Principe de la morphologie binaire

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

Image binaire



354



354

Modèle ensembliste : l'érosion

Sur une image binaire **l'érosion** diminue la taille des structures, et élimine les objets de taille inférieure à l'élément structurant. Elle permet de séparer les objets reliés.

Sur une image en niveaux de gris l'érosion diminue les niveaux de gris et augmente la taille des trous dans les objets.

$$\varepsilon_B(X) = \{x \in \Re^2; B_x \subseteq X\}$$

○ B_x élément structurant

Points x tels que l'élément structurant centré en x est inclus dans X

355



SORBONNE
UNIVERSITÉ

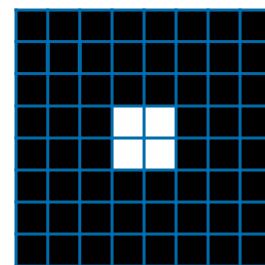
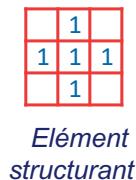
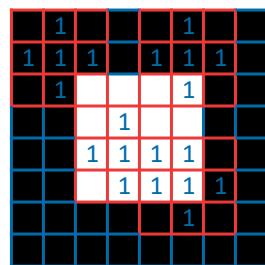
355

Erosion

$$\varepsilon_B(X) = \{x \in \Re^2; B_x \subseteq X\}$$

B_x = élément structurant

- Cette opération correspond à l'opérateur d'**inclusion** ou l'opérateur mathématique **min**
- Elle permet d'éroder les formes, c'est à dire diminuer leur taille. Les éléments de petites tailles disparaissent



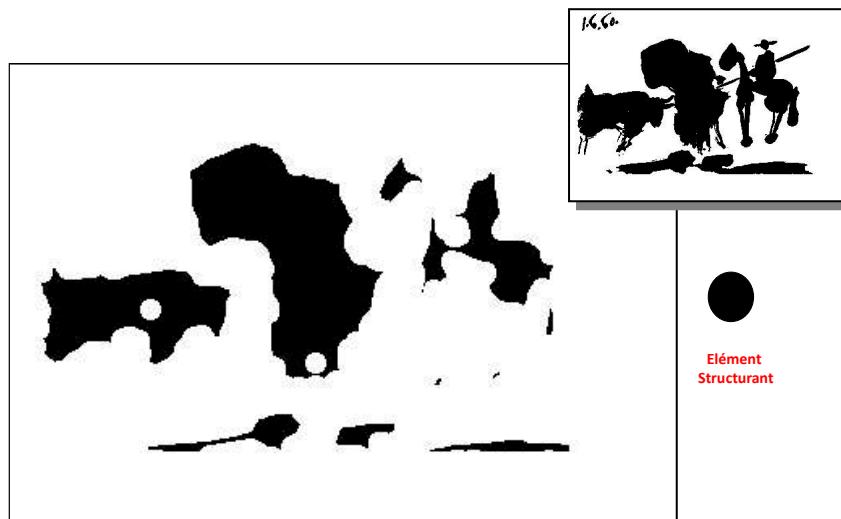
356

SORBONNE
UNIVERSITÉ

356

159

Modèle ensembliste : l'érosion



Pablo Picasso - Jeu de Cape III

357



357

Modèle ensembliste : la dilatation

La **dilatation** est l'inverse de l'érosion.

Elle grossit tous les objets en fonction de la taille de l'élément structurant.

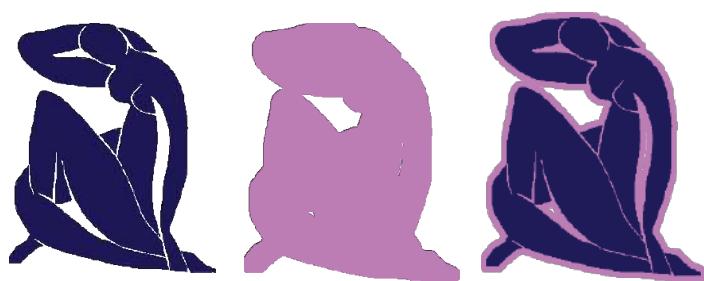
Les trous plus petits que l'élément structurant sont comblés

Les objets situés à une distance inférieure à la taille de l'élément sont fusionnés

$$\delta_B(X) = \{x \in \Re^2; B_x \cap X \neq \emptyset\}$$

B_x élément structurant

Points x tels que l'élément structurant centré a une intersection non nulle avec X



358



358

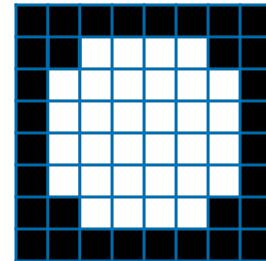
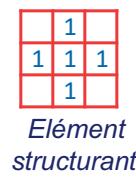
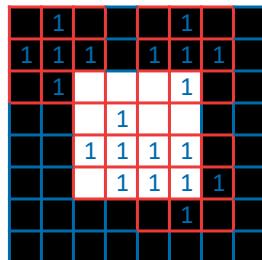
160

$$\delta_B(X) = \{x \in \Re^2; B_x \cap X \neq \emptyset\}$$

B_x élément structurant

Dilatation

- Cette opération correspond à l'opérateur d'**intersection** ou l'opérateur mathématique **max**
- Elle permet de dilater les formes, c'est à dire augmenter leur taille. Les « trous » de petites tailles disparaissent

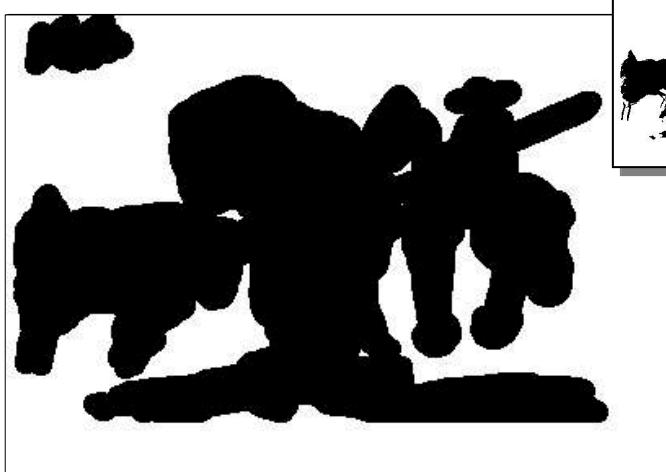


359



359

Modèle ensembliste : la dilatation



Elément Structurant

360

Pablo Picasso - Jeu de Cape III



360

161

Modèle ensembliste : dilatation et érosion fonctionnelles : niveaux de gris

Sur une image en **niveaux de gris** l'érosion consiste à donner à tout pixel la valeur minimale de l'image dans la fenêtre d'observation définie par l'élément structurant.

L'**érosion** tend à homogénéiser l'image (réduction des pics de niveaux de gris), à l'assombrir et à étaler le bord des objets les plus sombres.

$$\varepsilon_B^f(I)(p) = \inf \{I(s), s \in B_p\}$$

Sur une image en **niveaux de gris** la dilatation consiste à donner à tout pixel la valeur maximale de l'image dans la fenêtre d'observation définie par l'élément structurant.

La **dilatation** tend à homogénéiser l'image (épaisseur des pics), à l'éclaircir et à faire disparaître les objets sombres.

$$\delta_B^f(I)(p) = \sup \{I(s), s \in B_p\}$$

361



361

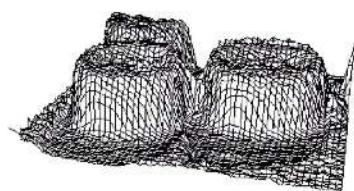
Erosion

Niveau de gris : principe et applications

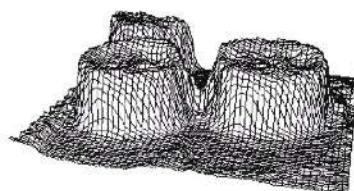
Gray Scale Erosion Example (continued)

Erosion effects can be seen in image surface plots.

Original bacteria image.



Eroded bacteria image.



The surface plots show the same results as the intensity images. The orientation of the surface plot is such that the edge of the surface plot closest to you is the left boundary of the image. Note how the thin region connecting the two cells has been diminished.

362



362

162

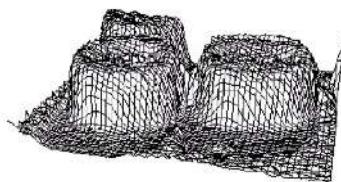
Dilatation

– Niveaux de gris : Principe et application

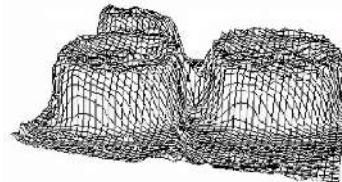
Gray Scale Dilation Example (continued)

Dilation effects can be seen in image surface plots.

Original bacteria image.



Dilated bacteria image.



The surface plots show the same results as the intensity images. The orientation of the surface plot is such that the edge of the surface plot closest to you is the left boundary of the image. Note how the thin region connecting the two cells has been enhanced.

363

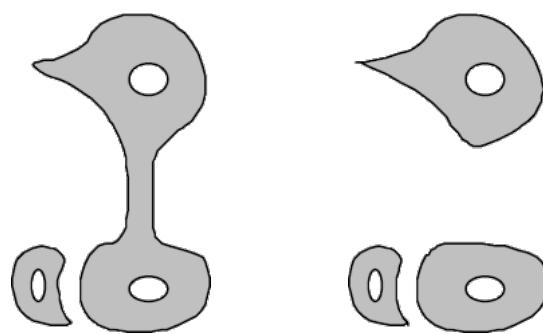


363

Modèle ensembliste : l'ouverture

Une **érosion** suivie d'une **dilatation** s'appelle une **ouverture**.

Comme le montre l'exemple, l'ouverture a pour propriété d'éliminer toutes les parties des objets qui ne peuvent pas contenir l'élément structurant.



364



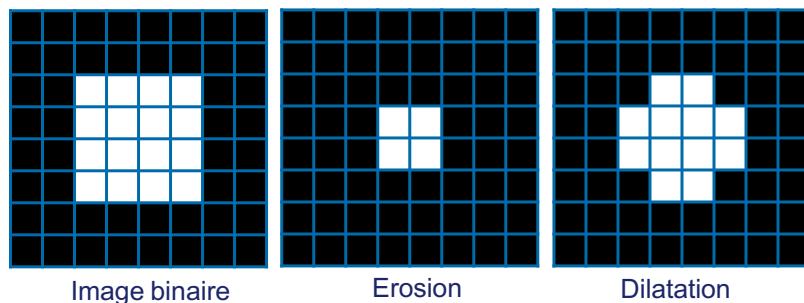
364

163

Ouverture

(érosion suivie d'une dilatation)

- Cette opération permet de supprimer les éléments de petites tailles comme le bruit tout en conservant la taille des formes



365



365

L'approche région

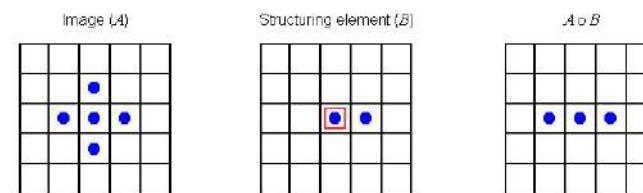
Ouverture

- C'est une érosion suivie d'une dilatation
- Cette opération permet de supprimer les éléments de petites tailles comme le bruit tout en conservant la taille des formes

The Open Operation



The open procedure is composed of an erosion followed by a dilation.



The open operation, \circ , is defined as an erosion followed a dilation.

$$A \circ B = (A \text{ eroded by } B) \oplus B$$

366



366

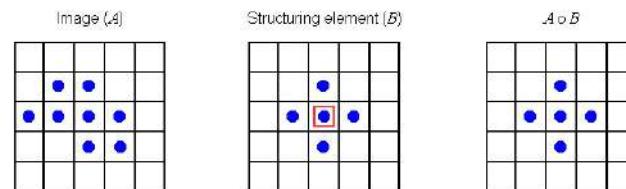
L'approche région

Ouverture – rôle :

- Lissage des contours

Open: Smoothing Contours

Use the open operation with a convex structuring element to smooth region boundaries.



Structuring elements that are convex cause a region with a jagged boundary to be replaced by a smoother one.

367



367

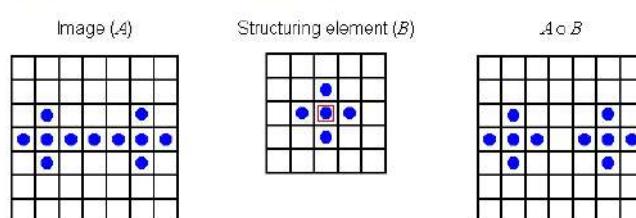
L'approche région

Ouverture – rôle :

- Lissage des contours
- Séparer les objets (trop) proches

Open: Breaking Narrows

Use the open operation with a convex structuring element to break an isthmus between blobs.



368



368

L'approche région

Ouverture – rôle :

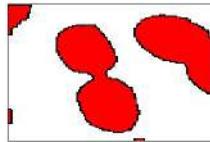
- Lissage des contours
- Séparer les objets (trop) proches

Open: Example

Use thresholding to produce a binary image from a gray scale image for the open operation.



This original bacteria image was passed through a threshold of 100. Pixels greater than 100 were set to true, all others were set to false. The open procedure was done on the resulting binary image.

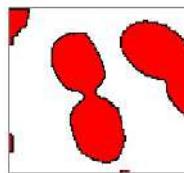


The new image after thresholding. The outlines show the region boundaries.

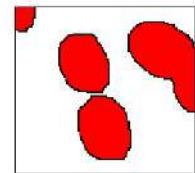
369

Open: Small vs. Large Structuring Elements

The size of the structuring element changes the results of the open operation.



This image shows the results after an open procedure using a small circular structuring element. This image is not significantly different from the original image. The narrow region is not even partially thinned since any thinning (eroding) is offset by region thickening (dilation).



This image shows the results of using a larger circular structuring element. The narrow region has successfully been broken, thereby separating the two bacteria cells. After the erosion there was no data joining the two regions. This means that there was no pixel data to dilate during the subsequent dilation.



369

L'approche région

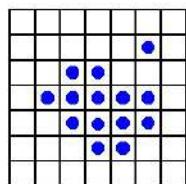
Ouverture – rôle :

- Lissage des contours
- Séparer les objets proches
- Eliminer le bruit

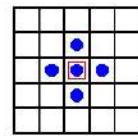
Open: Removing Noise

Use the open operation to remove noise from an image.

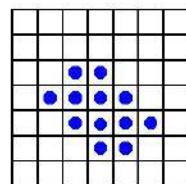
Image (A)



Structuring element (B)



$A \circ B$



The open operation also removes noise and other small regions whose size is smaller than that of the structuring element.



370

L'approche région

Ouverture

— Niveaux de gris - applications

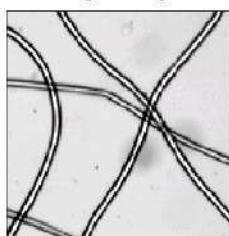
Gray Scale Open



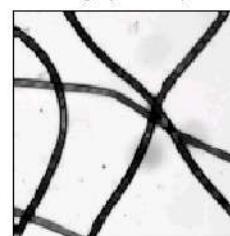
The gray scale open operation is an erosion followed by a dilation.

The gray scale open operation is defined as a gray scale erosion followed by a gray scale dilation. Just as with the binary open operation, the dilation uses the inverted structuring element. In many cases, the structuring element is symmetric, so the open operation can use same structuring element for both erosion and dilation.

Original image



After gray scale open



371

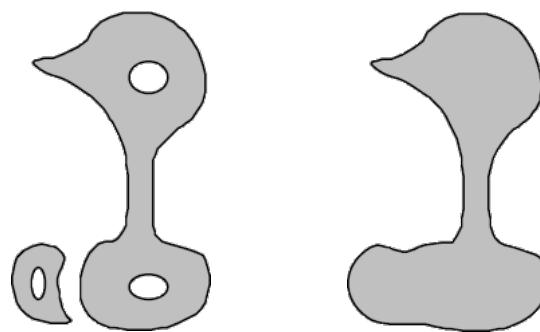
In the example, the original image contains a microscopic view of feather fibers.



371

Modèle ensembliste : la fermeture

- Une **dilatation** suivie d'une **érosion** s'appelle une **fermeture**.
- Comme le montre l'exemple, la fermeture a pour propriété de combler tout ce qui est de taille inférieure à l'élément structurant.



372

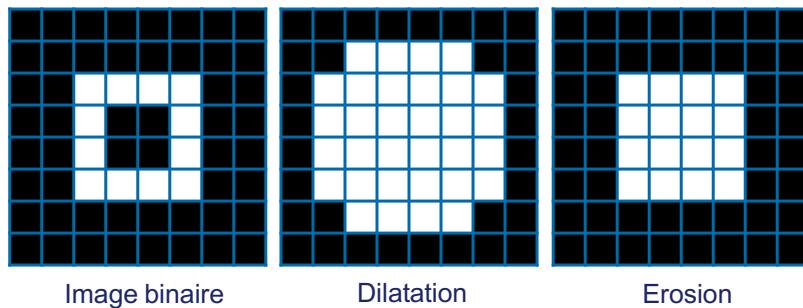


372

Fermeture

(**dilatation** suivie d'une **érosion**)

- Cette opération permet de supprimer les « trous » de petites tailles tout en conservant la taille des formes



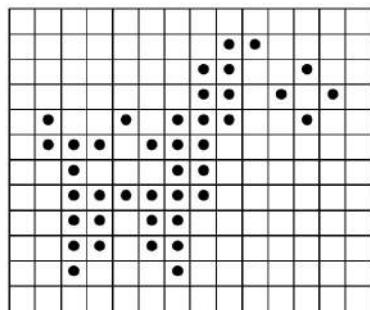
373



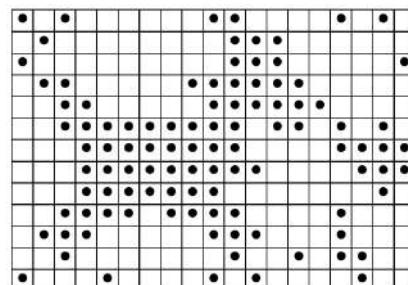
373

MM mini quizz

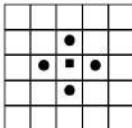
Fermeture / Closing



374

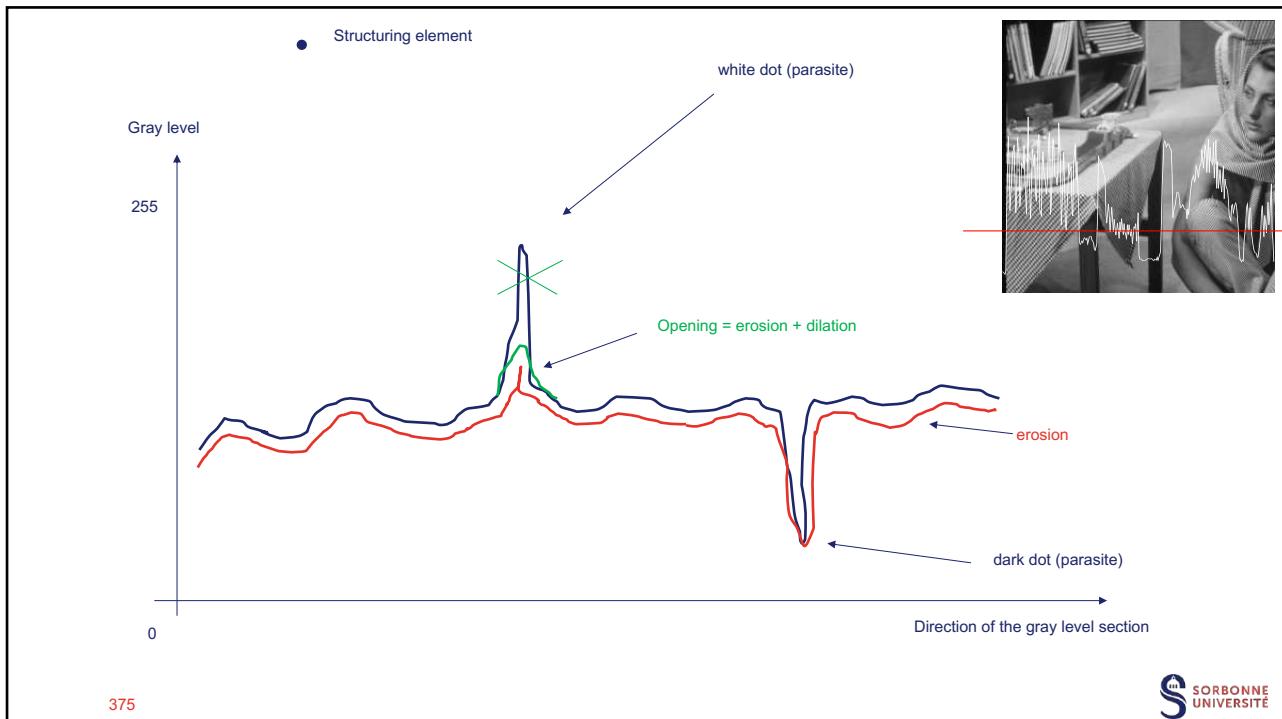


Ouverture / Opening

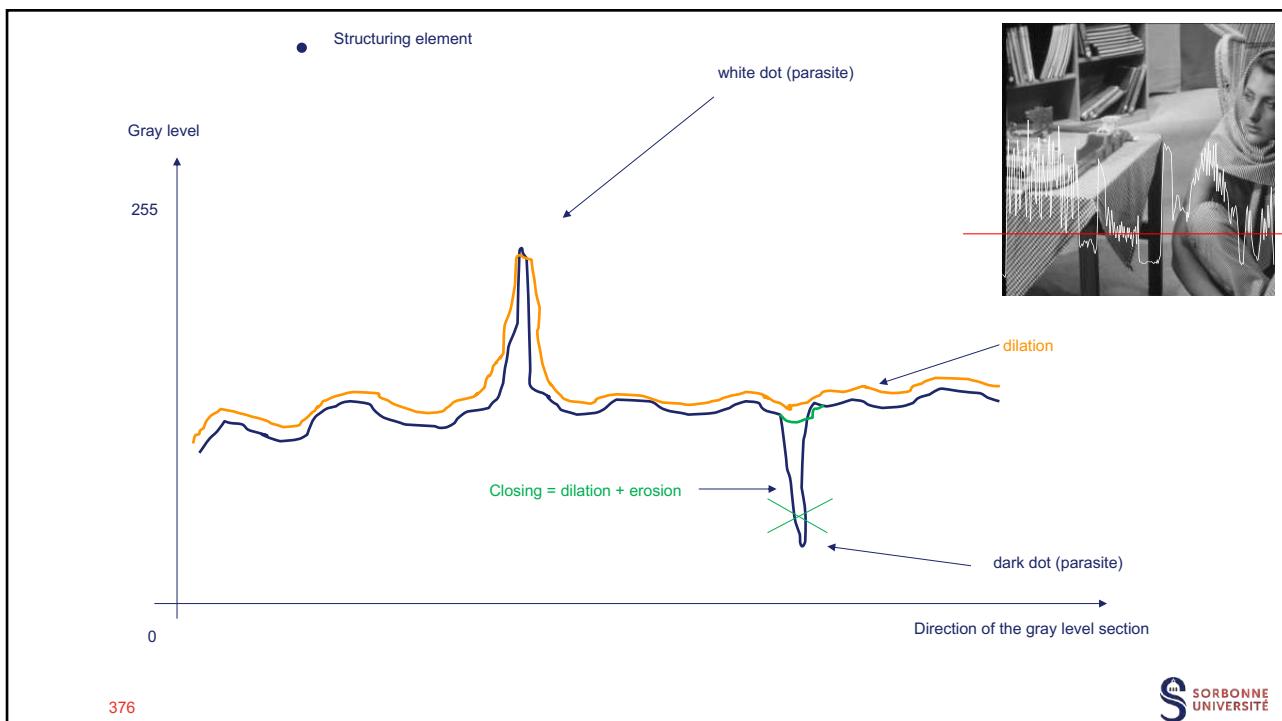


374

168



375

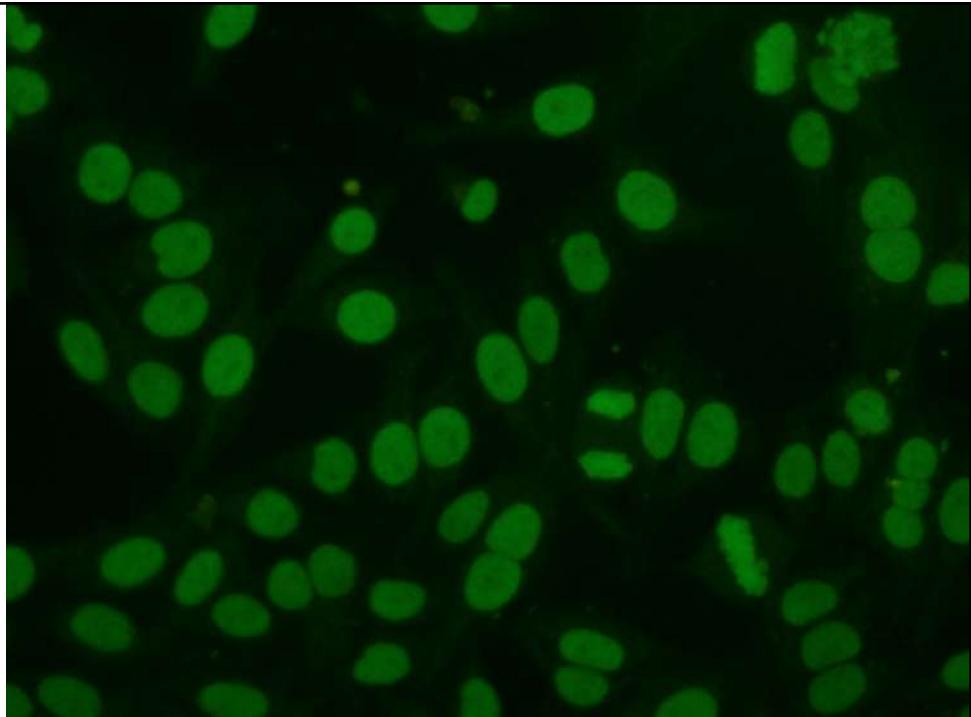


376

MM objectifs

Exemple
détection
de mitoses
projet Pitié
- Immuno

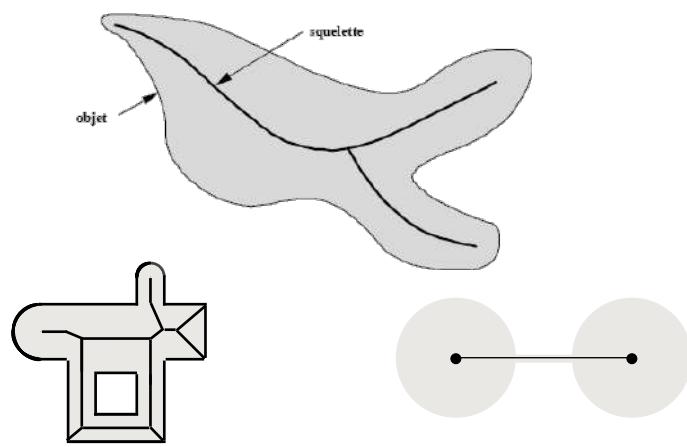
377



377

Modèle ensembliste : la squelettisation

- Le **squelette** d'une région est l'union des centres des boules maximales incluses dans cette région

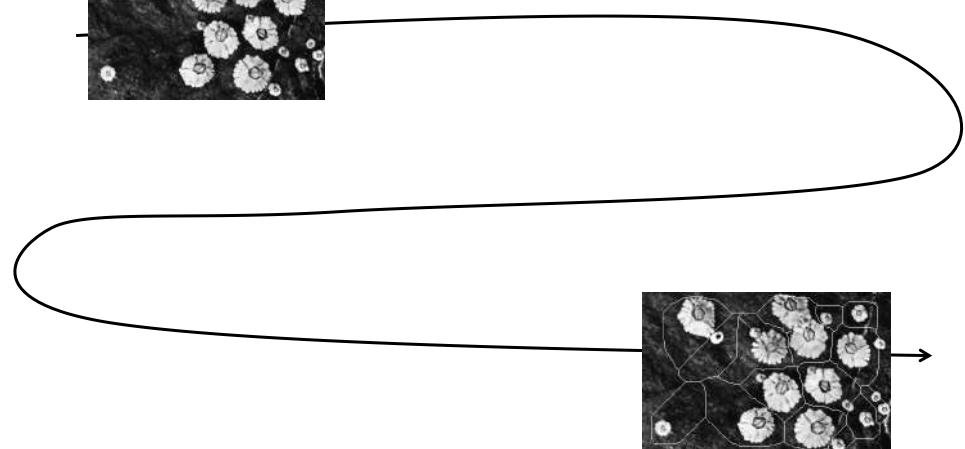


378



378

Quizz : trouvez l'enchaînement d'opérations :

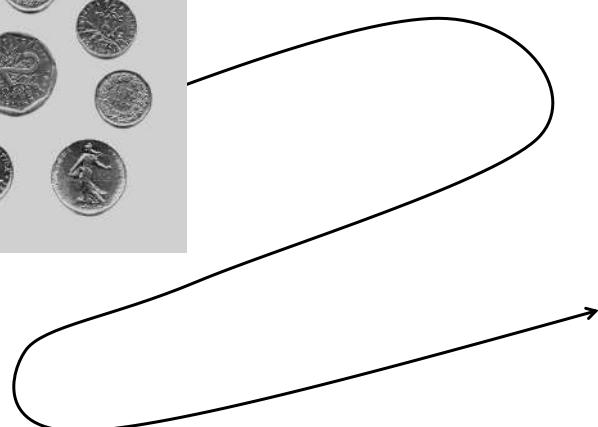


379

The Sorbonne Université logo, featuring a stylized blue 'S' with 'SORBONNE' and 'UNIVERSITÉ' written below it.

379

A vous de jouer !



380



380

171

f. Modèles discrets

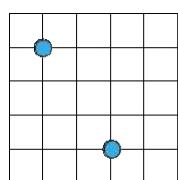
Traitement d'images



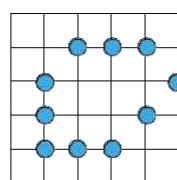
381

Modèle discret

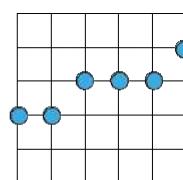
- La **géométrie discrète** est une discipline au moins aussi ancienne que le traitement d'images. Alors que le modèle différentiel considère les structures géométriques (courbes, surfaces, droites, ...) comme des approximations numériques de leurs homologues continues, ou que le modèle fréquentiel traduit la discréttisation en termes de perte d'information, le modèle discret, lui, intègre l'espace échantillonné comme cadre mathématique, et s'efforce de donner un cadre formel aux structures géométriques discrètes : définition, propriétés, théorèmes,...



Qu'est ce que la distance entre 2 points ?



Qu'est ce qu'un trou ?



Qu'est ce qu'une droite ?

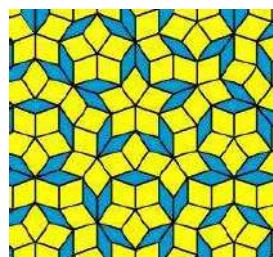
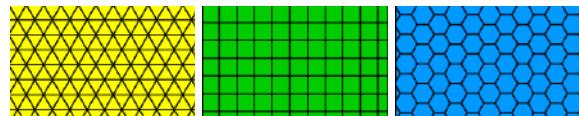
382



382

Modèle discret : pavage

- Un pavage du plan est une partition du plan en cellules élémentaires (pixels). Il existe trois pavages réguliers :
 - Triangulaire
 - Rectangulaire
 - Hexagonal
- ...et beaucoup d'irréguliers



383 Pavage apériodique de Penrose



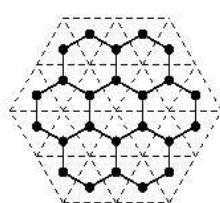
Pavage périodique d'Escher



383

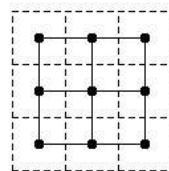
Modèle discret : maillage

- A tout pavage du plan on peut associer un *graphe* où les *sommets* (noeuds) représentent les cellules élémentaires, et où les *arêtes* représentent la relation d'*adjacence* entre les cellules (2 cellules sont adjacentes si elles ont un côté en commun). Un tel graphe est un *maillage* du plan.
- Les pavages et les mailles réguliers sont *duaux*

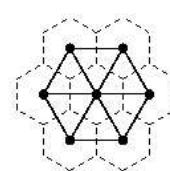


384

Pavage triangulaire
↓
Maillage hexagonal



Pavage carré
↓
Maillage carré



Pavage hexagonal
↓
Maillage triangulaire

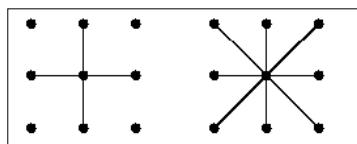


384

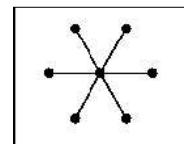
173

Modèle discret : connexité

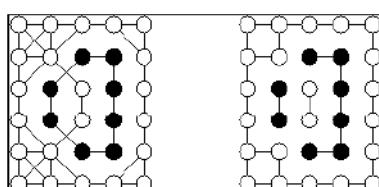
- Les relations topologiques dans les images discrètes sont définies à partir de la relation de connexité induite par le graphe du maillage



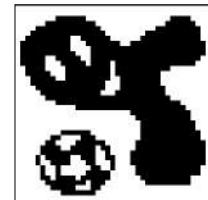
Maille carrée :
4-connectivité et 8-connectivité



Maille triangle :
6-connectivité



Combien de régions
connexes ?
• en 4-connectivité
• en 8-connectivité



SORBONNE
UNIVERSITÉ

385

385

Modèle discret : distance

Comment calculer la **distance** entre deux pixels $P(x_p, y_p)$ et $Q(x_q, y_q)$?

- **distance de Manhattan :**

$$d_1(P, Q) = |x_p - x_q| + |y_p - y_q|$$

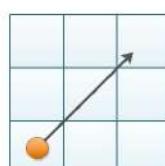
- **distance euclidienne :**

$$d_2(P, Q) = \sqrt{(x_p - x_q)^2 + (y_p - y_q)^2}$$

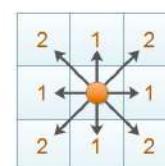
- **distance de l'échiquier :**

$$d_{\text{inf}}(P, Q) = \max(|x_p - x_q|, |y_p - y_q|)$$

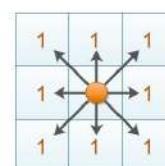
Euclidean Distance



Manhattan Distance



Chebyshev Distance



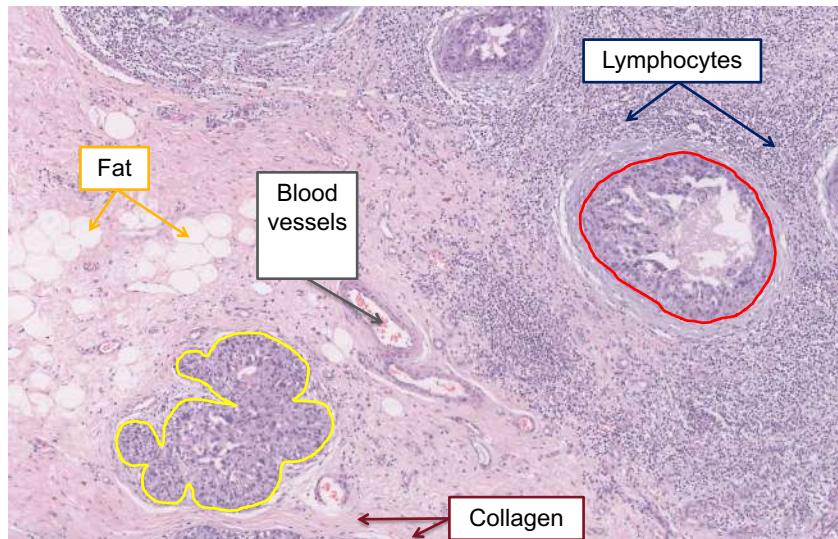
$$d_{\text{inf}}(P, Q) \leq d_2(P, Q) \leq d_1(P, Q)$$

386

SORBONNE
UNIVERSITÉ

386

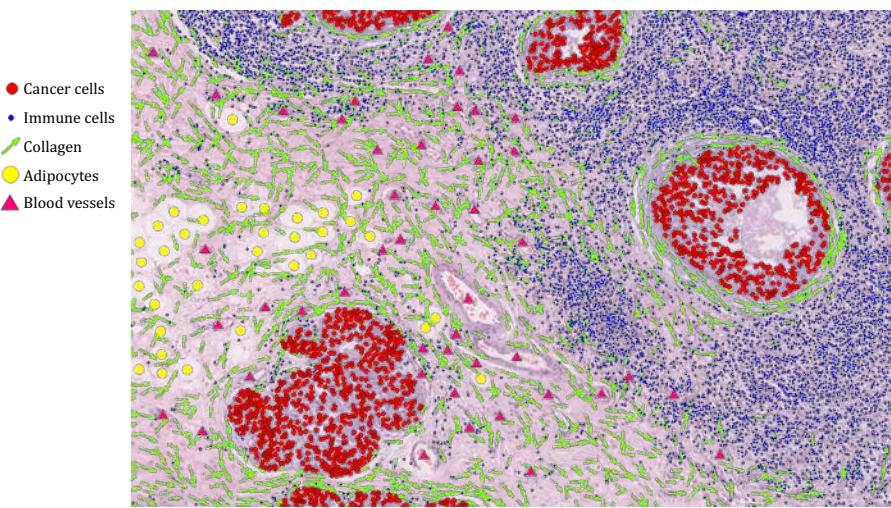
Tumor Micro-Environment Heterogeneity



387

387

Tumor Micro-Environment Heterogeneity



Spatial heterogeneity
in cancer ecosystem

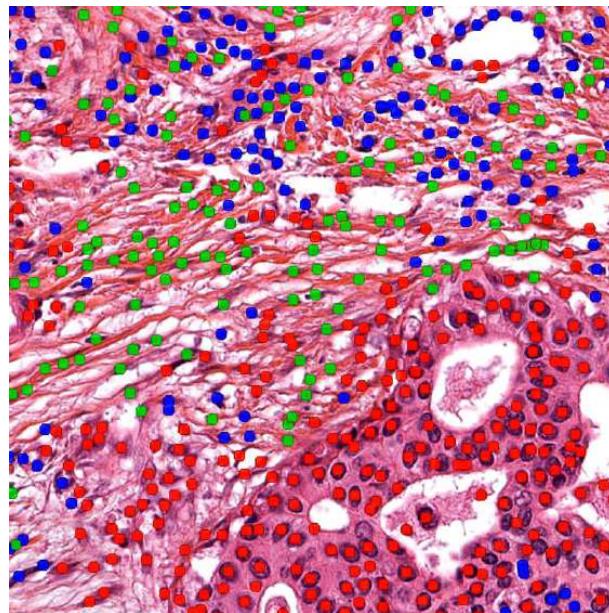
Quantitative analysis of spatial heterotypic
interactions in TME: Proximity/Adjacency,
Surroundedness/Enclosedness, Betweenness

388

388

175

Tumor Micro-Environment Heterogeneity



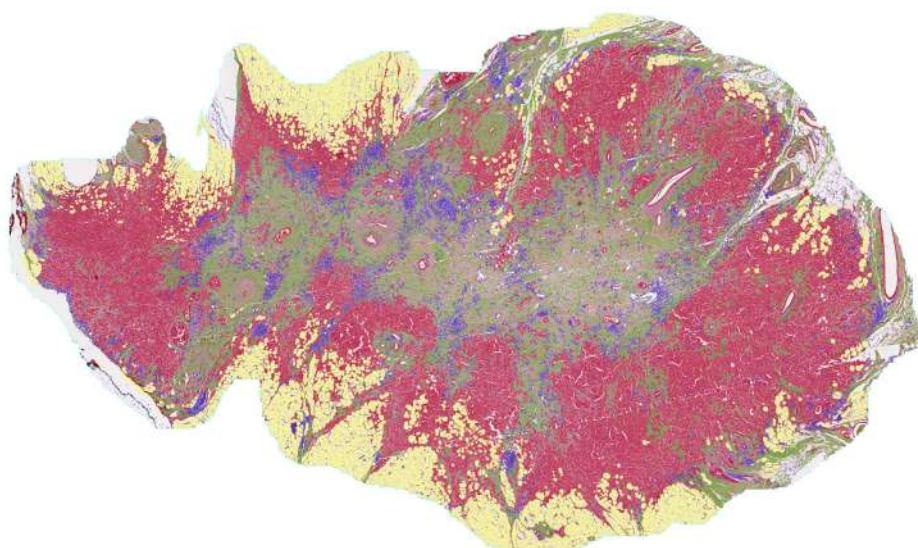
- Cancer cells
- Immune cells
- Collagen + Fibroblasts
- Adipocytes

389

 SORBONNE
UNIVERSITÉ

389

Nodes identification



- Cancer cells
- Immune cells
- Collagen + Fibroblasts
- Adipocytes

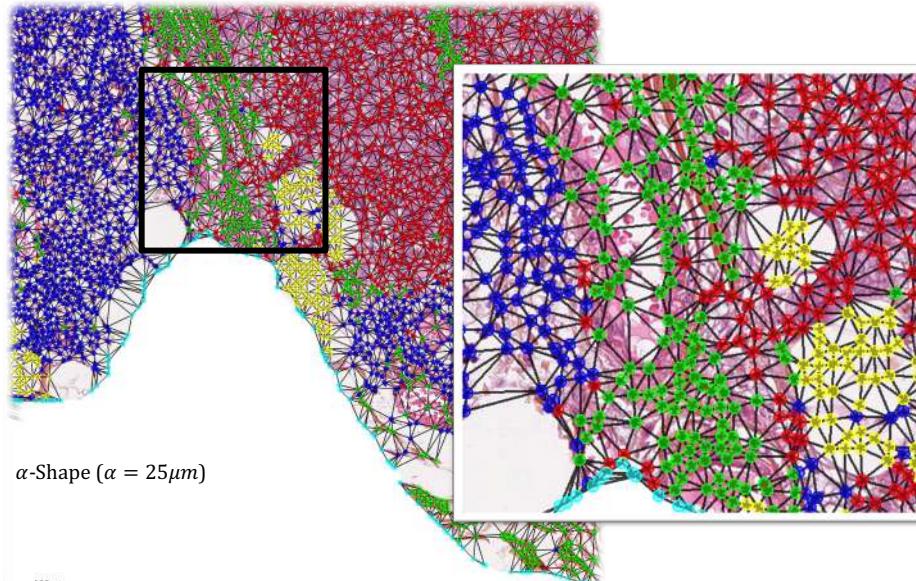
390

 SORBONNE
UNIVERSITÉ

390

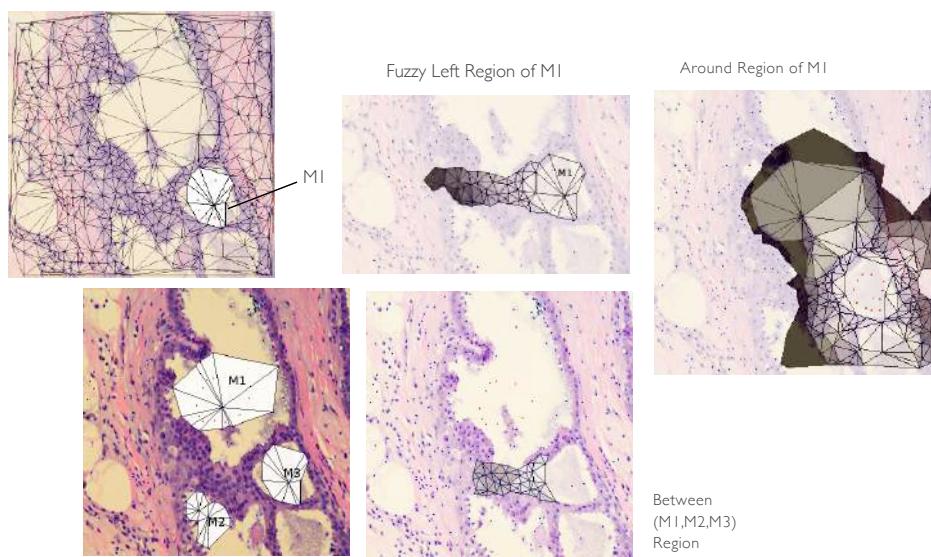
176

Spatial organization of histological structures



391

Architecture and Spatial Relationships



N. Lomenie, D. Racoceanu, Point set morphological filtering and semantic spatial configuration modeling: applications to microscopic image and bio-structure analysis, Pattern Recognition, 45(8), 2012

SORBONNE
UNIVERSITÉ

392

177

Spatial organization of histological structures

Sparsity:

-  Immune cell aggregates
-  Envelope immune cell aggregates

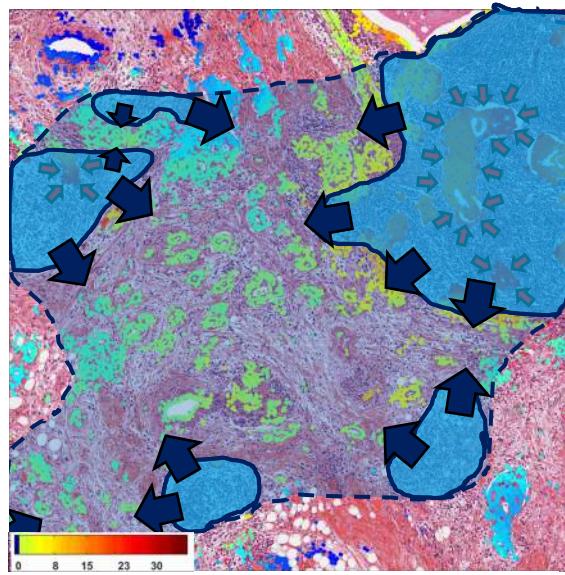
- Cancer cells
 - highly enclosed by immune cells
 - surrounded by immune cells
 - between distant immune cell aggregates
 - free from immune cells

Sparsity function:

$$\Gamma(G_X) = \frac{1}{N+1} \sum_{n=0}^N \zeta \circ \gamma_n(G_X)$$

$N = 5$

393

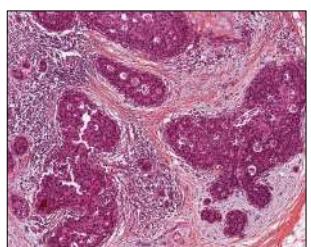


 SORBONNE UNIVERSITÉ

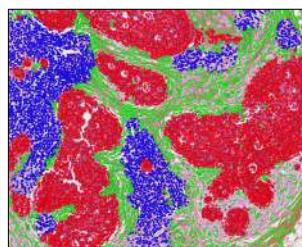
393

Interactions between tumor and immune system

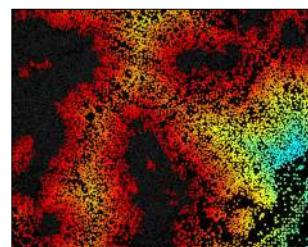
Morphological distance map:



Breast cancer tissue



Result of morphological filtering



Immune cell distance map

394

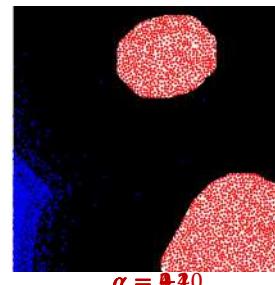
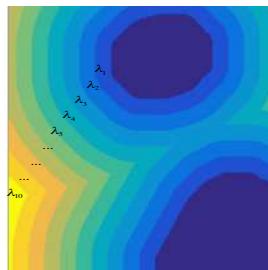
 SORBONNE UNIVERSITÉ

394

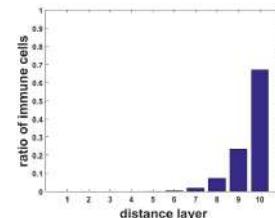
178

Tumor Microenvironment

□ A model of immune cells spatial distribution:



- Distance map
 - Quantization (parameter k)
 - Weight coefficients: $\lambda_i = \frac{i^{-\alpha}}{\sum_{i=1}^k i^{-\alpha}}$ (parameter α)
- $\alpha > 0$: immune cells close to tumor
 - $\alpha < 0$: immune cells distant from tumor
 - $\alpha = 0$: uniformly distributed



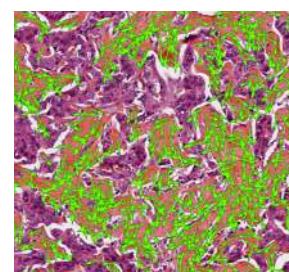
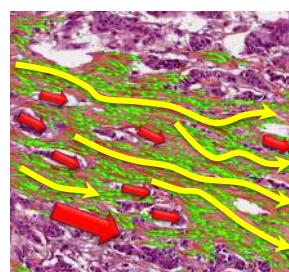
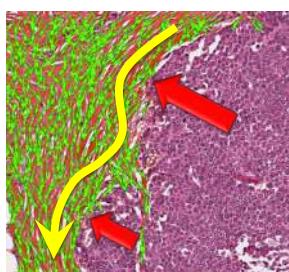
395



395

Tumor Microenvironment

□ Heterogeneity of the spatial organization of collagen fibers:



- Orientation
- Density
- Spatial proximity
- Surroundedness
- Stretchiness

396



396

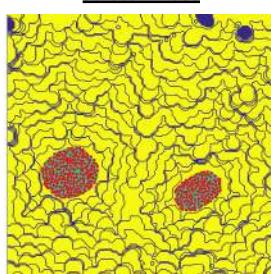
179

Tumor Microenvironment

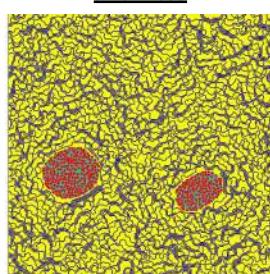
□ A model of collagen spatial arrangement:



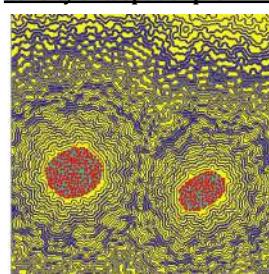
- Reshaping (parameter γ)
 - Distance map
 - Quantization (parameter k)
 - Dilation (parameter R)
- Morphological deformations of tumor patterns define various distance maps, which are quantized.

stretchiness

397

thickness

{1, 20, 30}

density and spatial proximity

{1, 2, 3, 5}



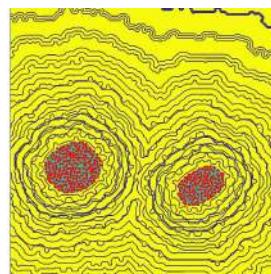
397

Tumor Microenvironment

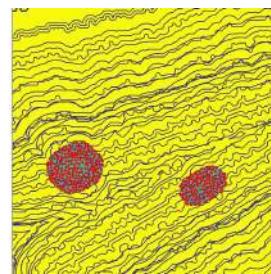
□ A model of collagen spatial arrangement:



- Reshaping (parameter γ)
 - Distance map
 - Quantization (parameter k)
 - Dilation (parameter R)
- Morphological deformations of tumor patterns define various distance maps, which are quantized.

Orientation relative to tumor

398



398

180

Tumor Patterns Modeling

DCIS IDC TC ILC

Random Points $\mathcal{P}(n_0, d_0)$
Tumor seeds

$\delta_{\Delta(L_0, \Theta_0)} \circ \delta_{R_0}$
Expanse

$\epsilon_{\Phi_{H_0}}$
Tubule formation

$\omega \downarrow$

Refinement

\otimes

	n_0	d_0
(a)	$\mathcal{N}_1(7, 2)$	800
(b)	$\mathcal{N}_1(40, 8)$	100
(c)	$\mathcal{N}_1(85, 5)$	40
(d)	$\mathcal{N}_1(100, 5)$	40

$\mathcal{N}_n(\mu, \sigma)$: n random values generated from a normal distribution
 $\mathcal{U}_n(a, b)$: n random values generated from a discrete uniform distribution

399

41 SORBONNE UNIVERSITÉ

399

Qualitative Results

Synthetic data

(a) DCIS (b) IDC (c) TC (d) ILC

Real data

DCIS IDC TC ILC

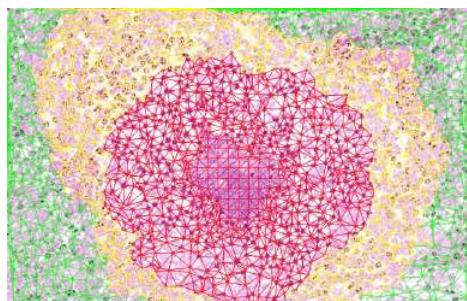
Ben Cheikh, B., Bor-Angelier, C., Racoceanu, D. (2017). A Model of Tumor Architecture and Spatial Interactions with the Microenvironment in Breast Carcinoma, SPIE Medical Imaging.

400

SORBONNE UNIVERSITÉ

400

Simulation in Digital Pathology ...



■ Non-hypoxic ■ Hypoxic ■ Necrotic

- Playing different scenarios
- Testing different profiling
- Discussing, analysing
- Educating
- Communicating
(e.g. with the patient)

- Data augmentation
- Creating synthetic data
(e.g. for DL)

- Finding correlations
- Simulating diff. therapies
- Prognosing
(one day, for sure ...)

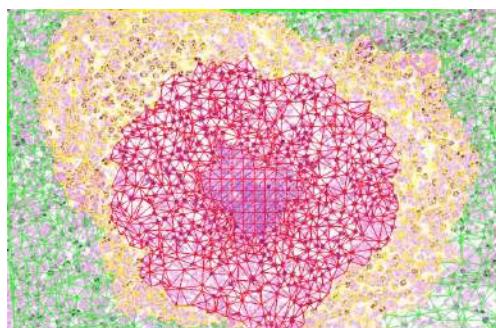


401

401

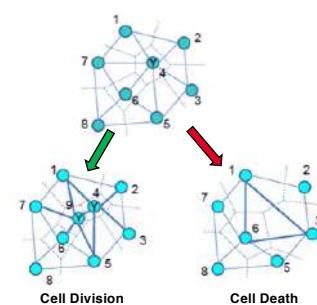
Spatial & Temporal Tumor Heterogeneity

Topological model framework for tumor cell growth simulations :



■ Non-hypoxic ■ Hypoxic ■ Necrotic

Graph Transform



Formalism that describes the dynamics of graphical structure

N. Loménie and D. Racoceanu, *Pattern Recognition*, 2012.

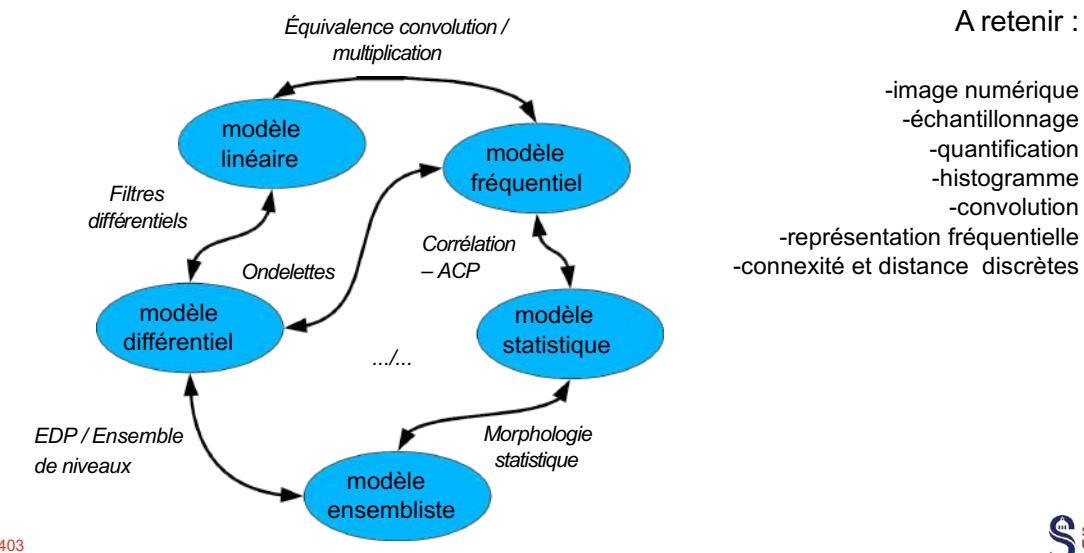
402

402



40

Conclusion : modèles fondamentaux



403