

# PROGRAMMATION CRÉATIVE

---

Florent Grélard

[florent.grelard@univ-lyon2.fr](mailto:florent.grelard@univ-lyon2.fr)

Master 2 Informatique | Conception et Intégration Multimédia, 2018-2019

# Sommaire

Objectifs et organisation du cours

Synthèse d'images

Analyse d'image

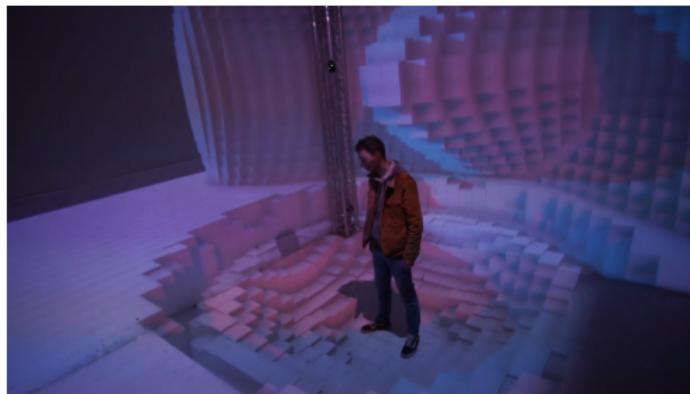
# Programmation créative

- **Interaction** entre l'œuvre et le spectateur
- Générer automatiquement des **images stylisées**
- Création d'**animations** (ex : Motion Design)

## Exemples : Interaction (1/3)

Mapping entre vidéo et bâtiments

Exemple : Studio THEORIZ basé à Lyon



Vidéo

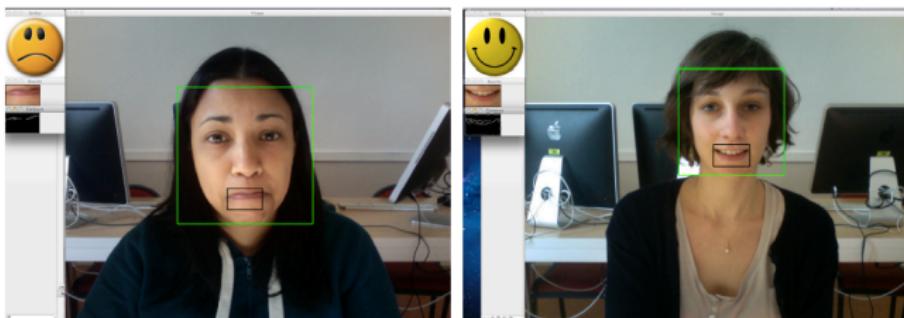
## Exemples : Interaction (2/3)

Incrustation d'images dans une vidéo (ex : Photobooth)



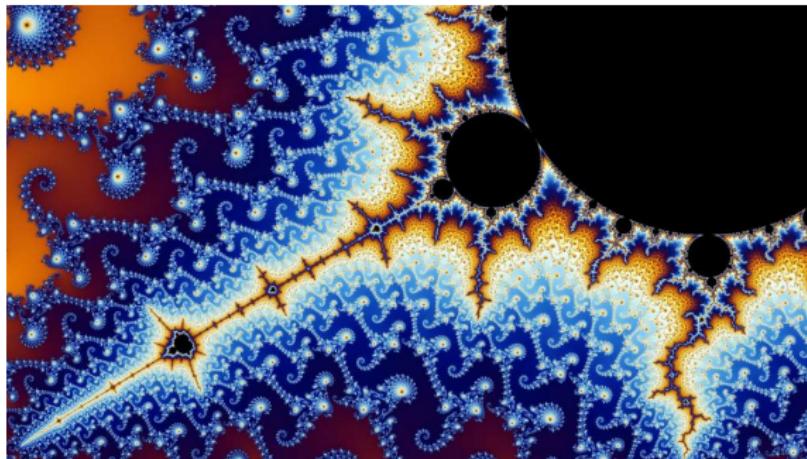
## Exemples : Interaction (3/3)

Projet étudiant : reconnaissance d'expressions du visage



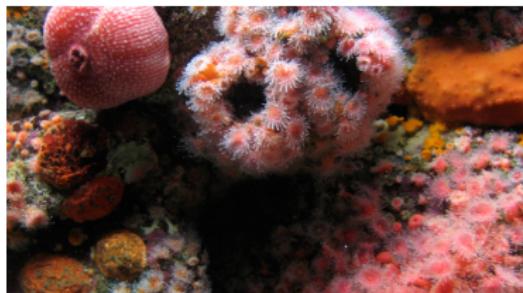
## Exemples : Images stylisées (1/2)

Fractales :



## Exemples : Images stylisées (2/2)

Pointillisme :



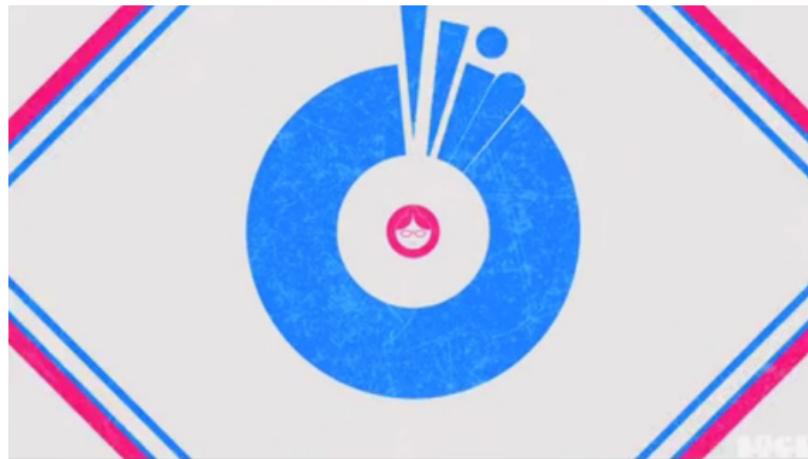
(a) Image originale



(b) Effet pointilliste

# Animations

Motion Design :



Vidéo

# Objectifs

Objectifs pédagogiques :

- Connaître les outils pour le traitement d'images et de vidéos
- Synthèse d'images
- Interagir avec un flux vidéo

Objectifs techniques :

- Manipulation de la bibliothèque Java *Processing* et son module *OpenCV* (fonctionnalités avancées)

## Organisation du cours (1/2)

Plan du cours :

- Cours (en décembre) :
  1. Synthèse d'images + Analyse d'images (opérations globales)
  2. Analyse d'images (opérations locales)
  3. Analyse de vidéos (1/2)
  4. Analyse de vidéos (2/2)
- Accompagnement et suivi de projet (en janvier)

## Organisation du cours (2/2)

Évaluation : Projet (4 séances)

Rendu avant le 30/01/2019 : document synthétique (maximum 3 pages) + code

# Sommaire

Objectifs et organisation du cours

Synthèse d'images

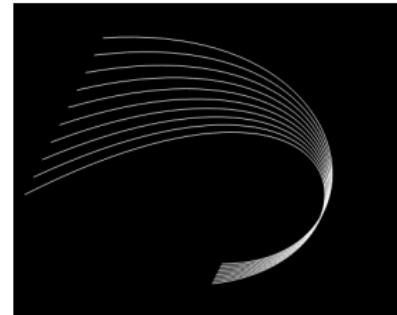
Analyse d'image

## Synthèse d'images : généralités (1/2)

**Synthèse d'images** : générer des images par la programmation, par la juxtaposition de primitives géométriques.

Intérêt :

- présenter/transmettre des notions (didactique)
- comprendre des notions (simulations)
- créer des images stylisées



# Synthèse d'images : généralités (2/2)

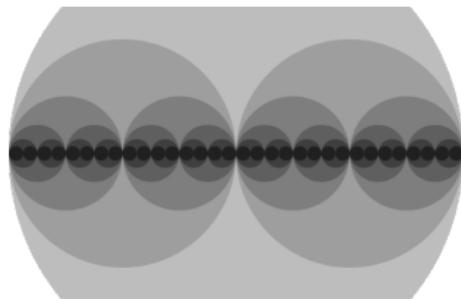
Les éléments d'un modèle :

1. des **primitives** : points, lignes, polygones 2D et 3D, cercles, courbes.
2. des **attributs** : styles, couleurs, textures.
3. des **relations de disposition** entre les composants du modèle : avant-plan, arrière-plan.

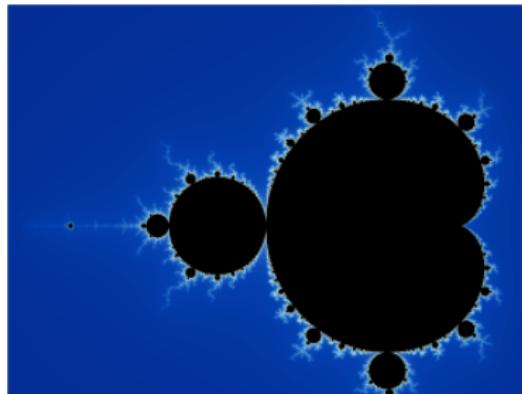
## Fractales (1/3)

Courbe dont la forme est invariante au niveau de zoom.

Répétition **récursive** de primitives géométriques (cf. *poupées russes*).



(a) Cercles à l'intérieur de cercles



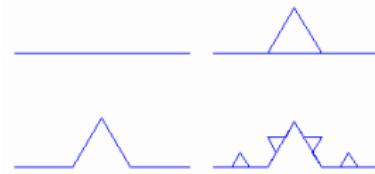
(b) Fractale de Mandelbrot

## Fractales (2/3)

**Exemple** : la courbe de Koch

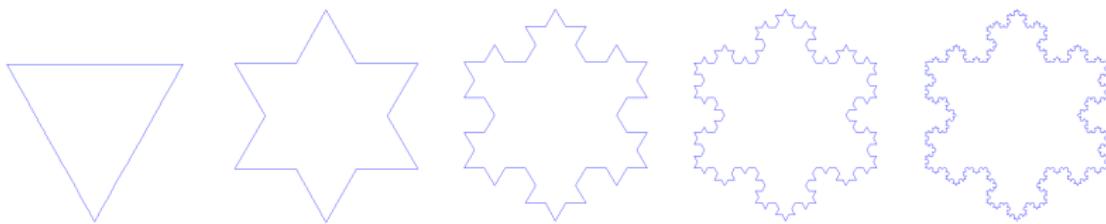
Chaque segment est modifié  
récursevement :

1. Division en trois segments de longueurs égales
2. Construction d'un triangle équilatéral ayant pour base le segment du milieu
3. Suppression du segment du milieu



## Fractales (3/3)

Exemple : la courbe de Koch



## Incrustation d'images

Ajout d'une primitive géométrique ou d'une image dans une image existante.



**Problème :** l'incrastation doit se faire automatiquement.

⇒ Besoin d'algorithmes d'**analyse d'images**.

Exemples : soustraction de fond, détection de formes : visages...

# Sommaire

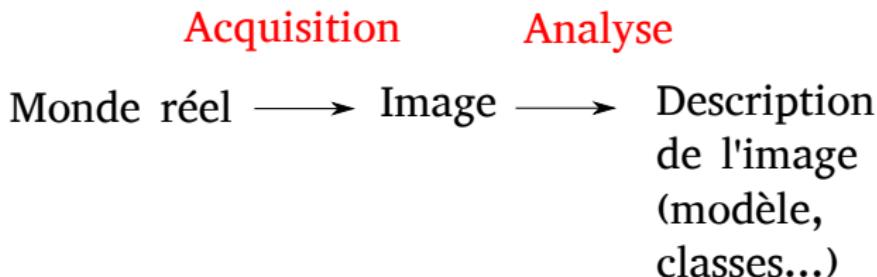
Objectifs et organisation du cours

Synthèse d'images

Analyse d'image

- Définitions
- Opérations globales sur l'image
- Opérations locales sur les images

# Analyse d'image



- Algorithmes imitant notre **perception visuelle**.
- Algorithmes généralement spécialisés pour un type **d'objet**

# Analyse d'image : exemples (1/2)

## 1. Segmentation

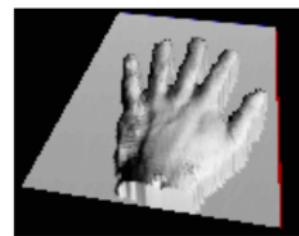
**Archéologie** : analyse de documents anciens  
 (DocCreator @ LaBRI)

GUIGNOL.

que tu n'y a pas la main ? est-ce qu'en fait de beau langage, celui-ci n'est pas tout à fait rupin et bien troussé ? — Quant aux belles manières, tu sauras, mon garçon, que Guignol les possède à fond , se trouvant, à cause de sa position sociale, en contact journalier avec la meilleure société qui se puisse voir en plein vent ; tels que marquis en bas âge, comtesses et baronnes en herbe, messdames leurs bonnes et mesdemoiselles leurs pouponnées. Il est vrai que c'est par ma voix que Polichinelle blague le commissaire ; mais c'est toujours en termes si bien choisis, que la censure n'y a jamais trouvé à redire. — Mais, dis-tu encore, je ne serais pas fâché de connaître un peu les bînettes de nos grands hommes modernes : est-ce que tu ne feras pas de biographies ? — Guignol ne pas faire de biographie ! il n'y aurait qu'un moyen de l'en empêcher : ce serait de lui couper les bras, les jambes et langue; encore serait-il capable, dans ce cas, d'écrire avec les dents ou le ventre : — cela s'est vu ! — Ainsi tu nous feras passer les grands hommes en revue ? — Pardieu ! Je te montrerai de face, de trois quarts, de profil, par derrière et par devant, depuis le talon jusqu'au sinciput, toutes les célébrités contemporaines tels que les Nisard et les Ponsard, les Chamfleury, Boyer, Jouvin, Villemessant, les Viard, Nadar, Aycard et Achard; les Malzon, German, Perraud et Vaucouleurs.

## 2. Vision 3D

**Comportement** : analyse de mouvements (shape from shading)



## Analyse d'image : exemples (2/2)

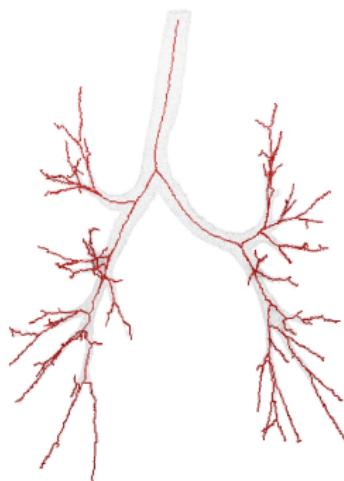
### 3. Reconnaissance de formes

**Botanique** : Reconnaissance de plantes (Folia @ LIRIS)



### 4. Mesures géométriques

**Médical** : Mesures géométriques sur des organes en 3D



# Chaîne de traitement

La caractérisation d'une image se fait en **plusieurs étapes** au sein d'une **chaîne de traitement** (ou *pipeline*).



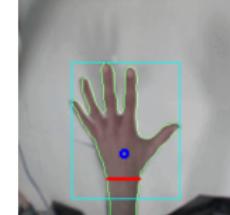
(a)



(b)



(c)



(d)

# Chaîne de traitement

Algorithmes :

- Entrée : une image
- Sortie : une image, des mesures, une catégorie...

Type de traitement :

1. **global** : concerne toute l'image (ex : seuillage sur histogramme)
2. **local** : sur une région de l'image (ex : filtres de convolution)

## Qu'est-ce qu'une image numérique ? (1/2)

- Image synthétisée ou obtenue par un dispositif de numérisation
- Processus de **discrétisation** du monde réel
- **Résolution** d'une image : nombre de pixels sur une longueur donnée

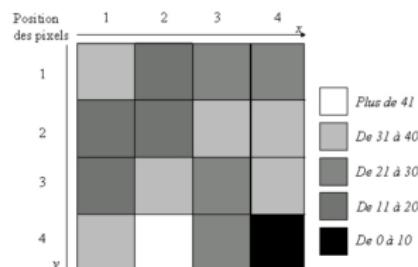


## Qu'est-ce qu'une image numérique ? (2/2)

Une **image** 2D est un tableau à deux dimensions composé de **pixels**.

Un **pixel** est l'élément unitaire d'une image, associé à :

- une position
- une "couleur" ou **intensité**

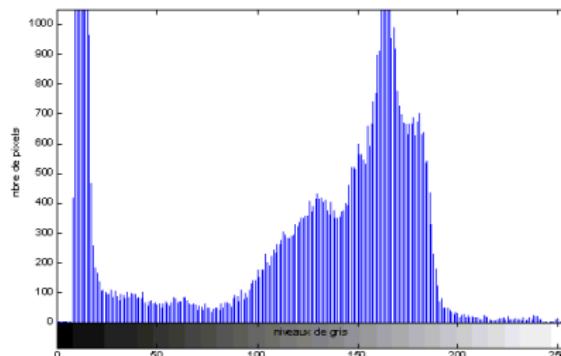


**NB :** Une image couleur contient 3 canaux : R (rouge), G (vert), B (bleu)

# OPÉRATIONS GLOBALES SUR L'IMAGE

## Histogramme

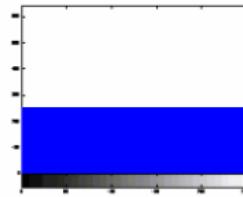
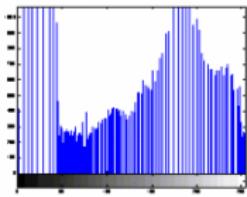
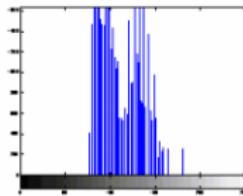
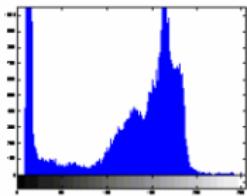
**Histogramme** : Graphique permettant de représenter la distribution des intensités des pixels d'une image (nombre de pixels pour chaque intensité).



# Histogramme

Intérêt :

- Etude de la **répartition des intensités** d'une image
- **Correction du contraste** et de l'échelle des couleurs pour des images sur-exposées ou sous-exposées



# Calcul de l'histogramme

Algorithme de calcul de l'histogramme :

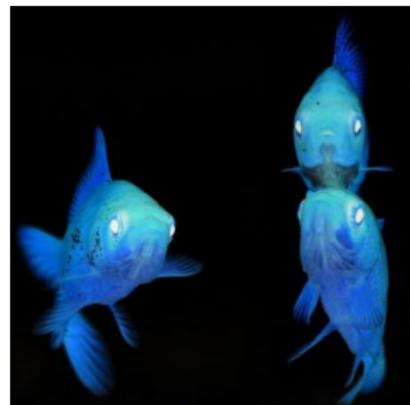
1. Initialisation d'un tableau  $T$  pouvant contenir 256 valeurs
2. Pour chaque pixel de l'image :
  - 2.1 Récupérer son intensité  $I$
  - 2.2 Incrémenter  $T[I]$  de 1

Souvent, l'histogramme est ensuite normalisé.

**Normalisation** : fréquence d'apparition (entre 0 et 1) pour chaque intensité  
⇒ traitement statistique de l'image

## Inversion d'histogramme

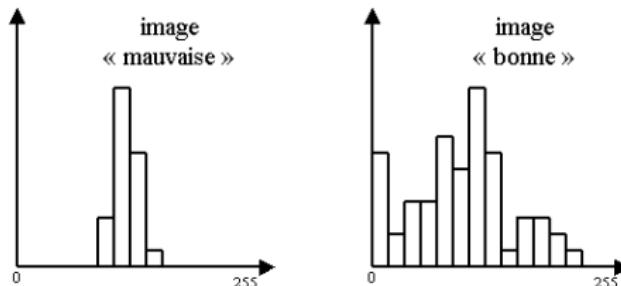
**Inversion** : intensités inversées par rapport à l'intensité maximale (255). Le résultat obtenu est un négatif de l'image originale.



## Égalisation d'histogramme (1/4)

**Objectif :** corriger des défauts d'illumination et de contraste dans l'image.

- Mauvaise image : histogramme “resserré”
- Bonne image : histogramme “étalé”



## Égalisation d'histogramme (2/4)

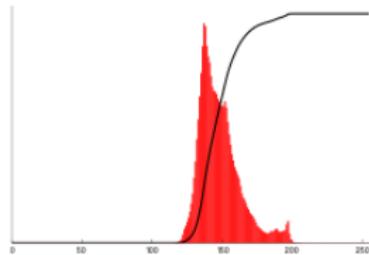
Égalisation de l'histogramme : permet de mieux répartir les intensités en visant à obtenir un histogramme “plat”.

Idée : histogramme cumulé  $T_{cumul}$  doit s'apparenter à une diagonale  $\Rightarrow$  proportionnalité entre l'intensité et la fréquence cumulée.

# Égalisation d'histogramme (3/4)

Algorithme d'égalisation :

1. Calcul de l'histogramme cumulé  $T_{\text{cumul}}$

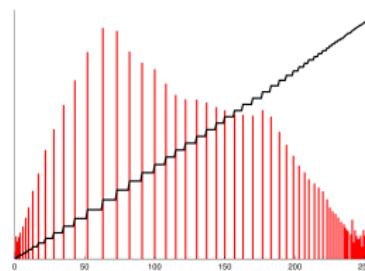
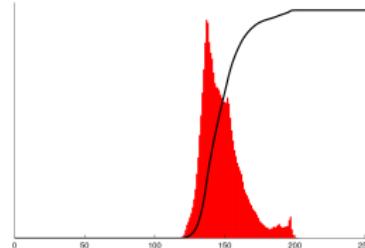


2. Pour chaque pixel d'intensité  $I$ , associer une nouvelle intensité  $I' = 255 * T_{\text{cumul}}[I]$

## Égalisation d'histogramme (4/4)

Égalisation :  $I' = 255 * T_{\text{cumul}}[I]$

Exemple :



## Seuillage sur histogramme (1/2)

**Seuillage simple** : les pixels prennent deux valeurs d'intensité : 0 ou 255, selon que leur intensité est inférieure ou supérieure à une certaine valeur (seuil).



(a) Image originale



(b) Seuil de 125

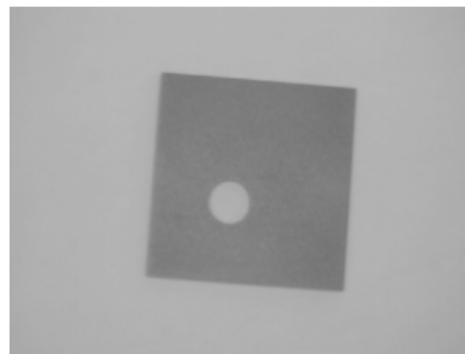


(c) Seuil de 200

## Seuillage sur histogramme (2/2)

- Résultat : image binaire → binarisation
- Permet de mettre en évidence des formes ou des objets dans une image
- Possibilité de faire un seuillage plus complexe : 2 seuils

Problème... Quel seuil permet de récupérer l'objet ci-dessous ?



# OPÉRATIONS LOCALES SUR L'IMAGE

# Opérations locales

Constat :

- Présence de motifs dans l'image
- Besoin de différencier certaines régions
- Suppression du **bruit**

**Bruit** : pixels d'intensités aberrantes (mauvaise luminosité, poussières, perturbations liées à d'autres appareils, etc.)



## Exemple de bruit

**Bruit poivre et sel** : attribue l'intensité 0 ou 255 à un nombre de pixels aléatoire

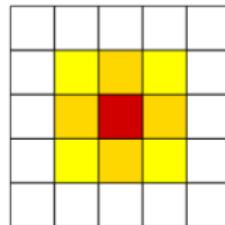
**Apparition** : lors d'une mauvaise transmission du signal, ou par numérisation



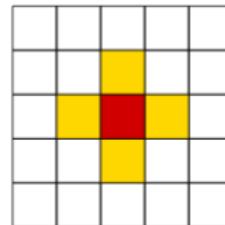
# Opérations locales

Traitement local : traiter l'image par régions

Voisinage d'un pixel : ensemble des pixels connexes au pixel, c'est-à-dire partageant un côté ou un sommet



8 voisins



4 voisins

Outils : opérateurs de morphologie mathématique, filtres.

## Effet pointilliste (1/2)

Idée : créer des régions dans l'image dont la couleur correspond à la couleur moyenne des pixels dans la région



(a) Image originale



(b) Effet pointilliste

## Effet pointilliste (2/2)

### Algorithme :

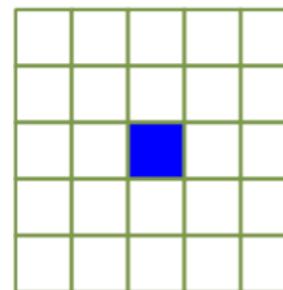
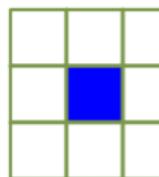
1.  $t$  = taille de la région
2. Pour chaque pixel  $p$  de l'image :
  - 2.1  $m_c$  = moyenne des intensités dans la région de taille  $t$
  - 2.2 Créer un cercle de centre  $p$  et de rayon  $t$  dont la couleur de remplissage est  $m_c$

## Opérateurs de morphologie mathématique (1/2)

Les opérateurs de morphologie mathématique permettent de :

- Supprimer du bruit sur des **images binaires**
- Extraire des contours sur des **images binaires**

## Opérateurs de morphologie mathématique (2/2)

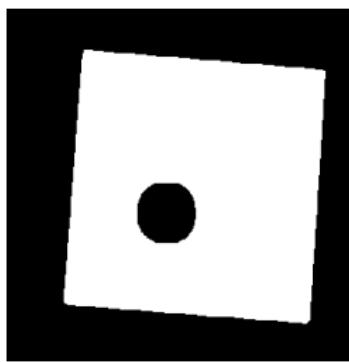


**Principe :** traiter l'image par des régions :

- Régions **binaires**, souvent carrées
- Positionnement de la région pour que son centre soit sur le pixel où on veut l'appliquer
- Renvoie une nouvelle intensité pour le pixel : blanc ou noir
- Parcours de l'ensemble de l'image

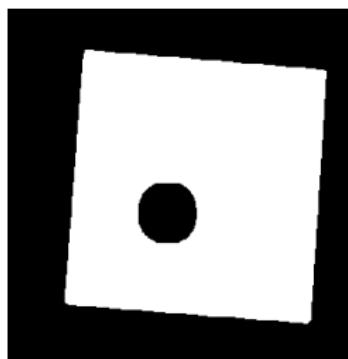
## Érosion

**Érosion** : un pixel est un pixel objet si la région centrée sur ce pixel ne contient que des pixels objet  
⇒ "rétrécit" l'objet



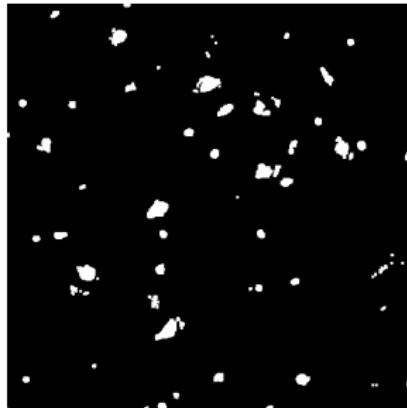
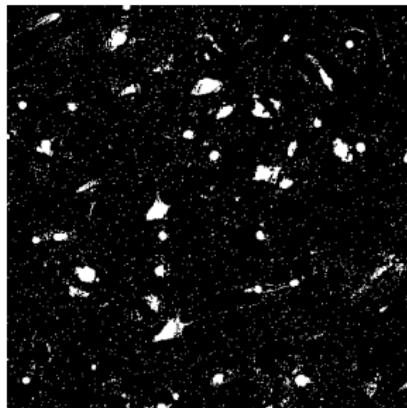
## Dilatation

**Dilatation** : un pixel est un pixel objet si la région centrée sur ce pixel contient au moins un pixel objet  
⇒ "épaissit" l'objet



## Ouverture

**Ouverture** : Érosion suivie d'une dilatation de l'image érodée



**Intérêt** : Supprime les petites structures. Permet de déconnecter des objets indépendants.

## Fermeture

**Fermeture** : Dilatation suivie d'une érosion de l'image dilatée

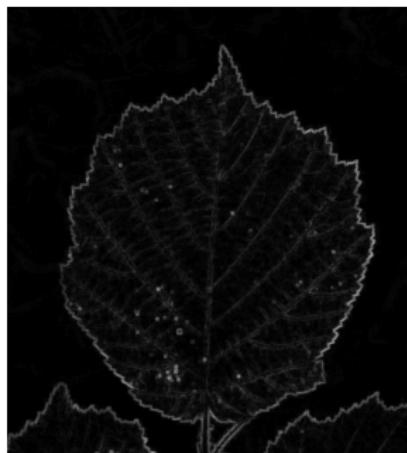
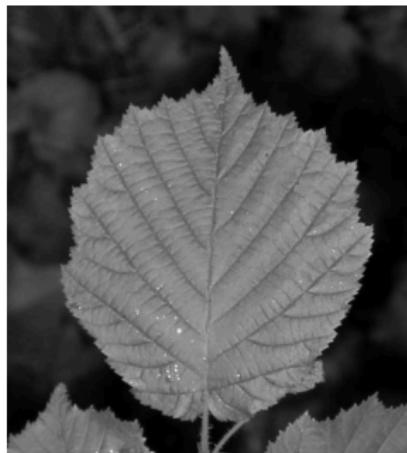


**Intérêt** : fusionne les structures proches, les trous sont comblés. Permet de recoller les composantes d'un objet.

## Gradient morphologique

**Gradient morphologique :** Différence pixel à pixel entre image dilatée et image érodée.

⇒ Donne des contours



## Filtres (1/5)

Les filtres permettent de :

- Améliorer la qualité (souvent visuelle) d'une image
- Supprimer le bruit

## Filtres (2/5)

**Filtres** : région contenant des coefficients. Ils modifient l'intensité d'un pixel en fonction de son voisinage

**Principe** : pondérer l'intensité du pixel par les pixels voisins

Importance de la **taille** du filtre (3x3, 5x5, 7x7...)

## Filtres (3/5)

**Calcul :** Pour calculer la nouvelle intensité d'un pixel obtenue par **convolution** avec un filtre :

1. On positionne le filtre sur l'image **centré** au pixel où on veut l'appliquer
2. Pour chaque pixel dans la région, on multiplie son intensité par le coefficient du filtre associé
3. On fait la **somme** de l'ensemble = nouvelle intensité

100	100	100	100	100
100	100	100	100	100
100	100	150	100	100
100	100	100	100	100
100	100	100	100	100

\*

0	-1	0
-1	5	-1
0	-1	0

=

100	100	100	100	100
100	100	100	100	100
100	100	350	100	100
100	100	100	100	100
100	100	100	100	100

## Filtres (4/5)

Algorithme :

1.  $f_t$  = filtre de taille t
2.  $R_t$  = région de taille t
3. Pour chaque pixel  $p$  de l'image :
  - 3.1  $I_{finale} = 0$
  - 3.2 Pour chaque pixel de coordonnées  $(x, y)$  dans la région  $R_t$  :
    - $I_{finale} +=$  intensité du pixel en  $(x, y)$  multipliée par le coefficient en  $(x, y)$  du filtre  $f_t$
  - 3.3  $I(p) = I_{finale}$

## Filtres (5/5)

Résumé en vidéo :



Convolution avec  
un filtre

## Filtre moyenneur (1/4)

**Filtre moyenneur** : L'intensité d'un pixel est remplacée par la **moyenne** de celles de ses voisins

Exemples :

0	1/5	0
1/5	1/5	1/5
0	1/5	0

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

## Filtre moyenneur (2/4)

Exemples :



(a) Image originale



(b) Image après filtre moyenneur  
 $3 \times 3$

## Filtre moyenneur (3/4)

Exemples avec du bruit poivre et sel :



## Filtre moyenneur (4/4)

Avantages :

- Permet d'éliminer les intensités avec une fréquence importante (bruit poivre et sel par exemple)

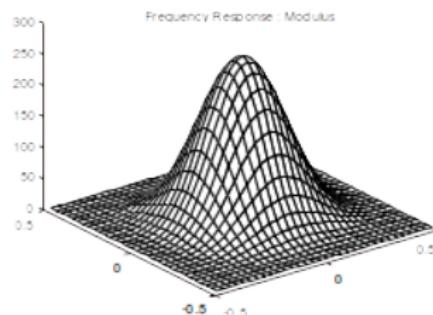
Inconvénients :

- Élimine les détails de l'image
- L'image résultante est plus floue

## Filtre gaussien (1/3)

Gaussienne : Fonction mathématique  $\Rightarrow$  courbe en "cloche".

Permet de lisser l'image (flou)



1	1	2	2	2	1	1
1	2	2	4	2	2	1
2	2	4	8	4	2	2
2	4	8	16	8	4	2
2	2	4	8	4	2	2
1	2	2	4	2	2	1
1	1	2	2	2	1	1

## Filtre gaussien (2/3)

Exemples avec du bruit poivre et sel :



## Filtre gaussien (3/3)

Avantages :

- Supprime le bruit dans les parties homogènes de l'image
- Dégrade moins les détails que le filtre moyenneur

Inconvénients :

- Conserve une partie du bruit dans l'image

## Filtre médian (1/3)

**Filtre médian** : Remplacer l'intensité d'un pixel par la l'intensité **médiane** de ses voisins

**Intérêt** : Même si plusieurs pixels voisins sont bruités, on peut corriger le pixel courant

## Filtre médian (2/3)



## Filtre médian (3/3)

Avantages :

- Bons résultats bruit poivre et sel
- Élimine moins les détails de l'image que le filtre moyenneur

Inconvénients :

- Lissage : même des pixels corrects peuvent être modifiés
- Coûteux : temps de calcul important car tri des voisins pour chaque pixel

## Bilan

Limites des opérations locales :

- Images complexes : on ne peut pas toujours réduire l'analyse à des traitements locaux.
- Les *a priori* dont découle l'analyse sont parfois erronés

⇒ Méthodes d'apprentissage (**ex** : laisser le soin à l'ordinateur de trouver les coefficients d'un filtre pour détecter une forme).