

Outils d'imagerie pour la robotique

Analyse d'images

Florent Grélard
florent.grelard@u-bordeaux.fr

TS341 | Option Robotique, 2022 – 2023

Sommaire

Objectifs et organisation

Image numérique

Analyse d'images

Organisation

- **Bases de l'analyse d'image** : 3 séances (cours + TP)
↳ Florent Grélard
- **Intelligence artificielle et apprentissage** : 2 séances
(cours + TP)
↳ Michaël Clément
- **Projet** : 4 séances + 1 séance de soutenance
↳ Michaël Clément + Sébastien Delpeuch

Outils d'imagerie pour la robotique

Objectifs pédagogiques :

- Caractérisation d'une image
- Notion de couleur
- "Boîte à outils" d'analyse d'images et de vidéos

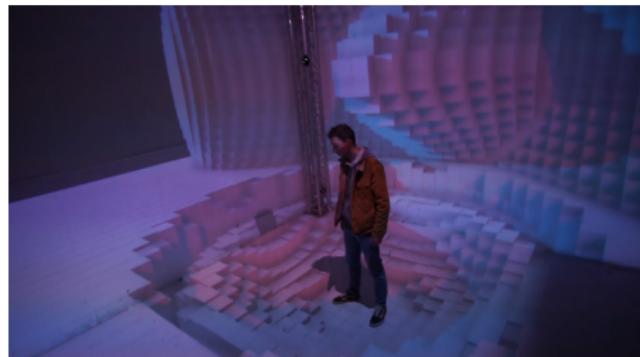
Objectifs techniques :

Manipulation de la bibliothèque OpenCV en Python.

Exemples : Interaction (1/2)

Mapping entre vidéo et bâtiments

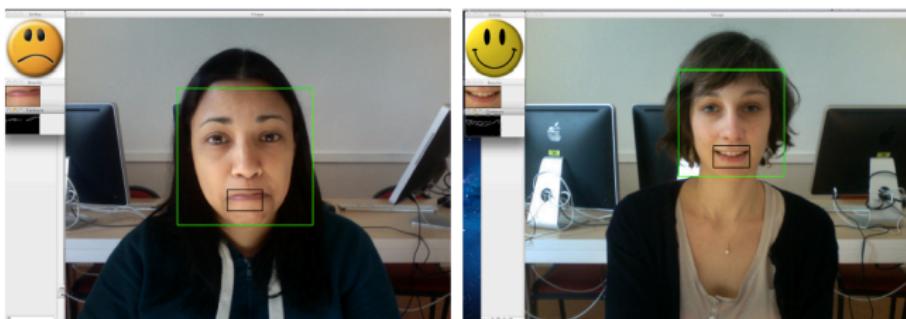
Exemple : Studio THEORIZ basé à Lyon



Vidéo

Exemples : Interaction (2/2)

Projet étudiant : reconnaissance d'expressions du visage



Exemples : Images stylisées

Pointillisme :



(a) Image originale



(b) Effet pointilliste

Exemples : analyse d'images (1/2)

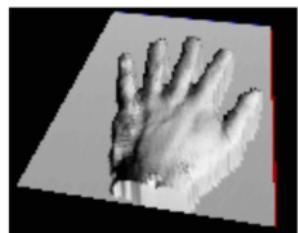
1. Segmentation

Patrimoine : analyse de documents anciens
(DocCreator @ LaBRI)

2.
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 tressé ? — Quant aux belles manières, tu sauras, mon garçon, que Guignol les possède à fond¹, se trouvant, à cause de sa position sociale, en contact journalier avec la meilleure société, qui se puise voir en plein vent; tels que marquis en bas âge, comtesses et baronnes en herbe, mesdames leurs bonnes et mesdemoiselles leurs poupees. 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 binettes 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, Féerie, Villeneuve, les Vives, Nodier, Ampère, etc., etc.

2. Vision 3D

Comportement : analyse de mouvements (shape from shading)



Exemples : analyse d'images (2/2)

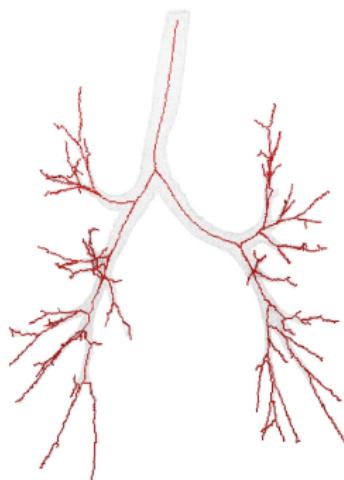
3. Reconnaissance de formes

Botanique : Reconnaissance de plantes (Folia @ LIRIS)



4. Mesures géométriques

Médical : Mesures géométriques sur des organes en 3D



Exemples : Incrustation

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



Problème : l'incrustation doit se faire automatiquement.

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

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

Sommaire

Objectifs et organisation

Image numérique

- Définitions
- Espaces couleurs

Analyse d'images

Qu'est-ce qu'une image numérique ? (1/3)

- 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

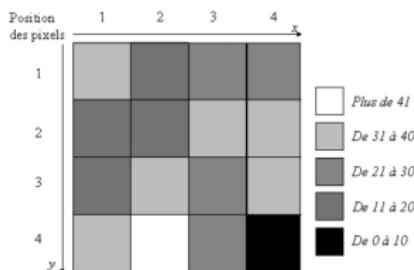


Qu'est-ce qu'une image numérique ? (2/3)

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)

Qu'est-ce qu'une image numérique ? (3/3)

Une image numérique $I: D \subset \mathbb{Z}^n \rightarrow [0, C_1] \times \cdots \times [0, C_p]$.

D : **domaine** = grille à coordonnées entières

p : nombre de **canaux**

C_i : nombre de **niveaux** pour ce canal

Ex : [0-255] pour une image 8-bit

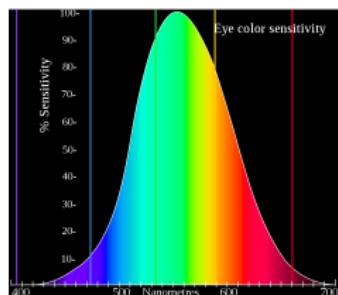
ESPACES COULEURS

Perception des couleurs

La perception des couleurs allie des phénomènes **physiques** et **biologiques**.

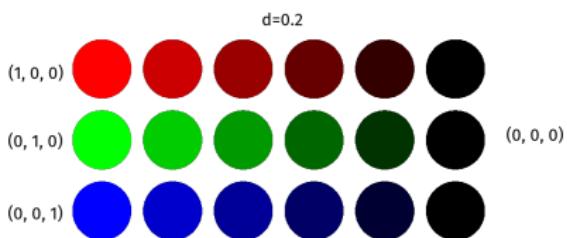
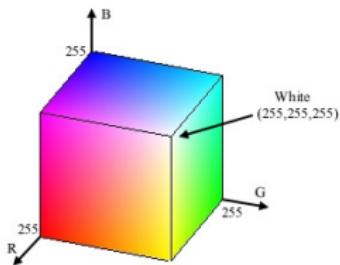
Cellules de la rétine :

- **Bâtonnets** : information d'intensité : **luminance**
- **Cônes** : information de couleur : **chrominance**



Espace RGB

Espace 3D :



La **luminance** est donnée par : $L = \frac{R + G + B}{3}$.

Perception de luminance différente en fonction du canal.

Luminance vs chrominance

Intensités identiques pour chaque canal :



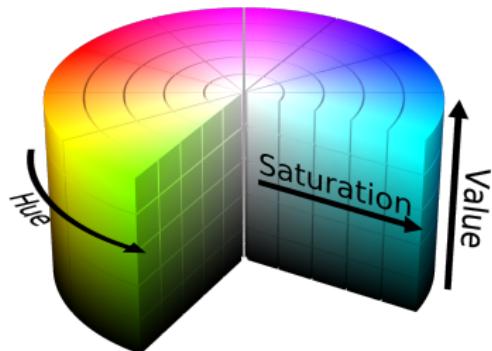
⇒ **Problème** : l'information est répartie dans les 3 canaux.
↳ peu intuitif, éloigné de la perception.

Espace HSV

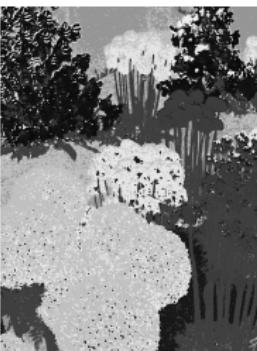
Définition d'un espace couleur plus intuitif.

3 canaux, coordonnées polaires :

- Teinte (hue) : couleur
- Saturation : taux de pureté de la couleur
- Valeur : intensité lumineuse de la couleur



Espace HSV



(a) Teinte



(b) Saturation



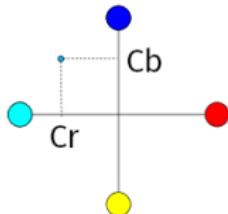
(c) Valeur

Espace YCbCr

YCbCr :

- Un canal pour la luminance : Y
- Deux canaux pour la chrominance : Cb et Cr

Cb et Cr correspondent au contraste Bleu/Jaune et Rouge/Cyan, respectivement.



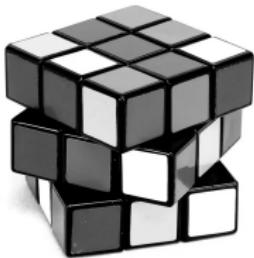
Espace YCbCr

Les composantes sont obtenues par les formules :

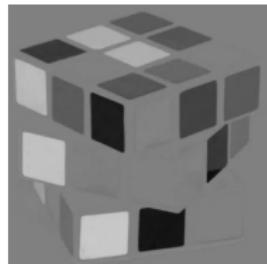
$$\begin{cases} Y = 0.299R + 0.587G + 0.114B \\ Cb = 0.564 * (B - Y) + 128 \\ Cr = 0.713 * (R - Y) + 128 \end{cases} \quad (1)$$



(a) Y



(b) Cb



(c) Cr

Sommaire

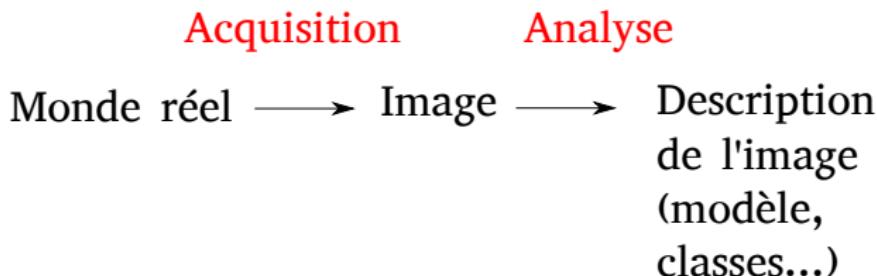
Objectifs et organisation

Image numérique

Analyse d'images

- Opérations globales
- Opérations locales
- Détection de contours

Analyse d'image (1/2)

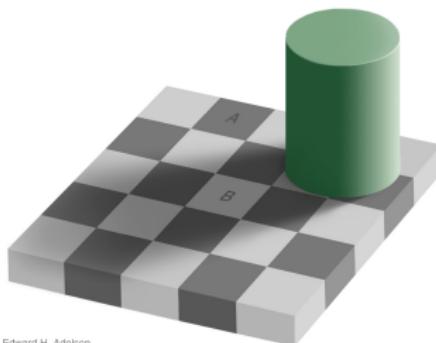


- Algorithmes imitant notre perception visuelle.
- Algorithmes généralement spécialisés pour un type d'objet

Analyse d'image (2/2)

Problème : la vision humaine est **complexe**.

Transposer les mécanismes de la vision à un ordinateur : **pas de solution unique**.

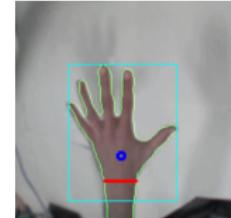


Edward H. Adelson

Chaîne de traitement

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

Exemple :



Chaîne de traitement

Algorithmes :

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

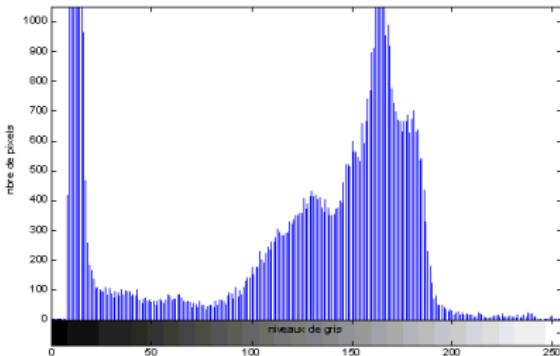
Type de traitement :

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

OPÉRATIONS GLOBALES SUR L'IMAGE

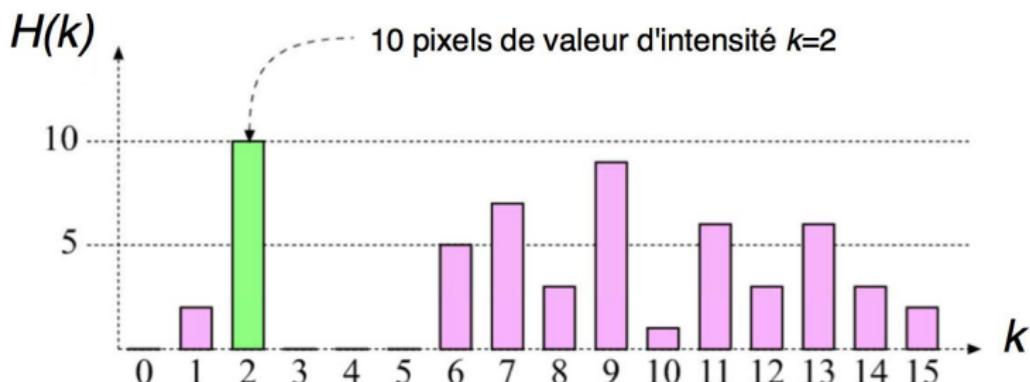
Histogramme

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



Histogramme

Histogramme $H(k) = \text{card}(\{I(i,j) = k, (i,j) \in W \times L\})$

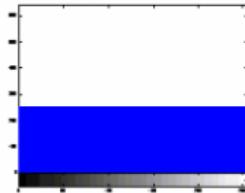
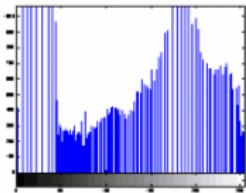
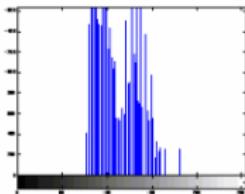
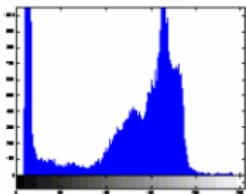


$H(k)$	0	2	10	0	0	0	5	7	3	9	1	6	3	6	3	2
k	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Histogramme

Intérêt :

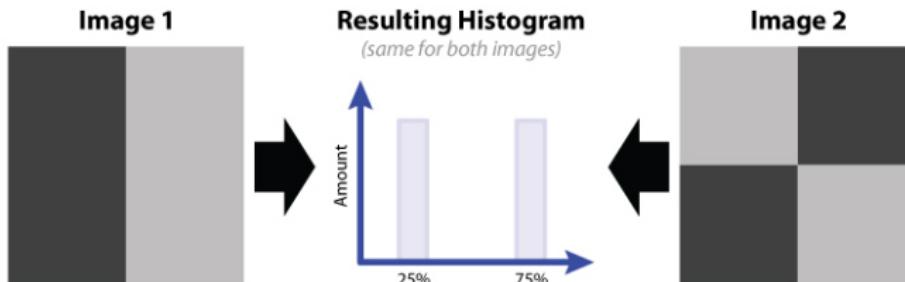
- Etude de la **répartition des intensités** d'une image
- **Correction du contraste** et de l'échelle des couleurs pour des images sur-exposées ou sous-exposées



Histogramme

Précautions :

- H ne reflète pas l'**information spatiale**
- 2 images différentes peuvent avoir le même histogramme
- On ne peut reconstruire l'image à partir de H .



Calcul de l'histogramme

Algorithme de calcul de l'histogramme :

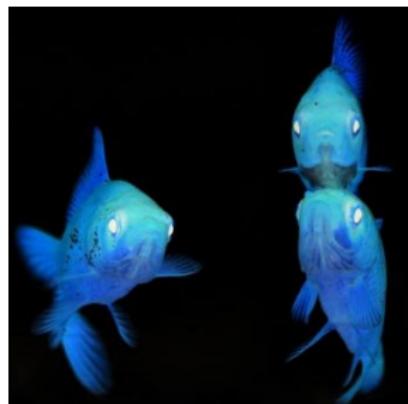
1. Initialisation d'un tableau H pouvant contenir 256 valeurs
2. Pour chaque pixel de l'image :
 - 2.1 Récupérer son intensité I
 - 2.2 Incrémenter $H[I]$ de 1

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”

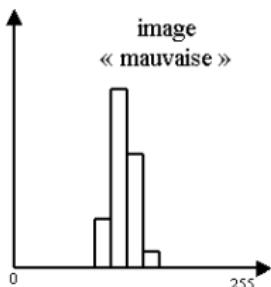


image
« mauvaise »

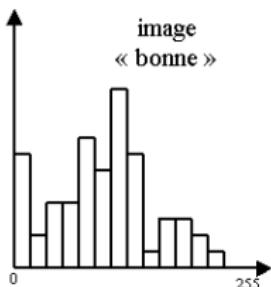


image
« bonne »

Égalisation d'histogramme (2/4)

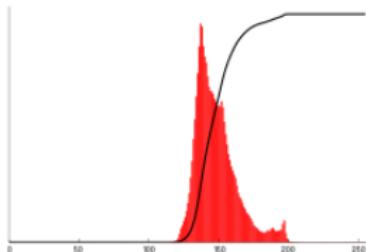
Égalisation de l'histogramme : permet de mieux répartir les intensités en visant à obtenir un histogramme “plat”.

Idée : histogramme cumulé H_{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 H_{cumul}

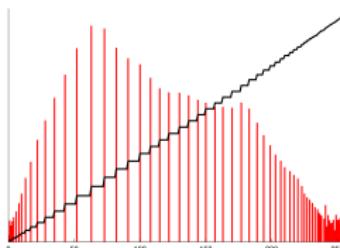
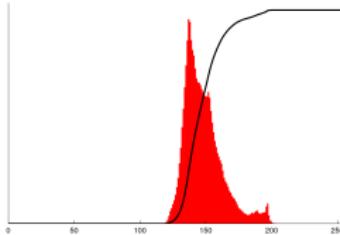


2. Pour chaque pixel d'intensité I , associer une nouvelle intensité $I' = 255 * H_{cumul}[I]$

Égalisation d'histogramme (4/4)

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

Exemple :



Seuillage sur histogramme (1/4)

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 (2/4)

Seuillage d'Otsu : trouver un seuil T automatiquement

- **A priori** : distribution bi-modale.
- **Idée** : obtenir deux classes séparées au maximum.
- **Principe** : Minimisation variance intra-classe \leftrightarrow Maximisation variance inter-classe

Seuillage sur histogramme (3/4)

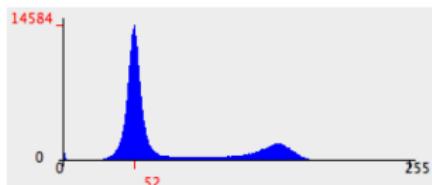
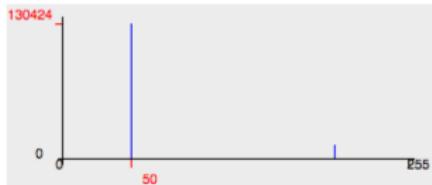
Seuillage d'Otsu :



$$T = 119$$



$$T = 101$$

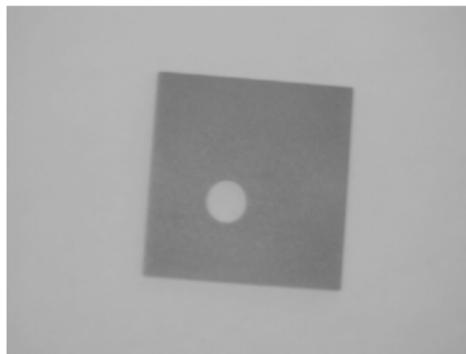


הוורן וטומסן ממכר לטלפון
טמיון אל-תונש אש את איזו ברכ
ב-אזרח ריבול, תבנה מאטעמות
באותם מכיל כפירה השניות
ז' ולי מעת שנים רמייט מכך
רמאנתרוא מכיר לך לאתני
עווראות מלאדר' כעוי יהוה נא
ים אהיקת את משפטיהם תשובה
שם אהם ישבעם על ראיין
הרארן פיריה ואכלתפ לשבען וו

Seuillage sur histogramme (4/4)

- **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

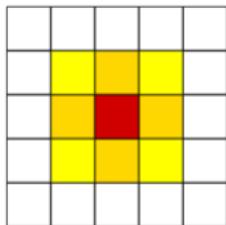
Apparition : lors d'une mauvaise transmission du signal, ou par numérisation



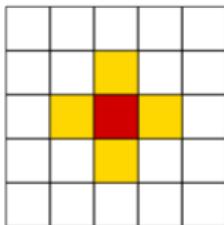
Opérations locales

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)

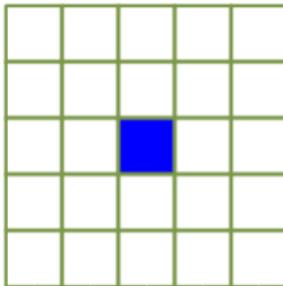
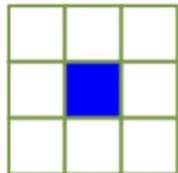
Les opérateurs de morphologie mathématique permettent de :

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

Image binaire : **Objet(s)** : pixels blancs (=255)

Arrière-plan : pixels noirs (=0)

Opérateurs de morphologie mathématique (2/2)

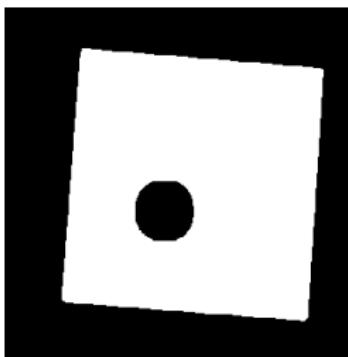


Principe : traiter l'image par des régions :

- 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

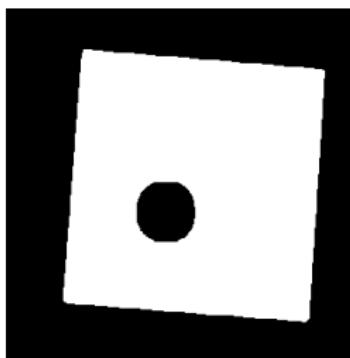
Érosion

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



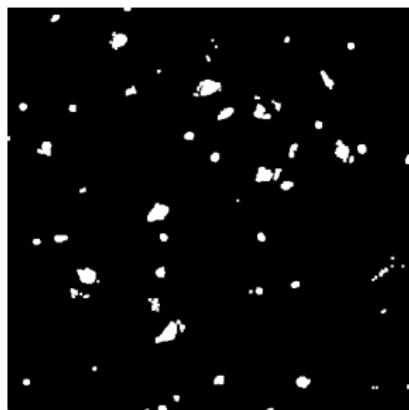
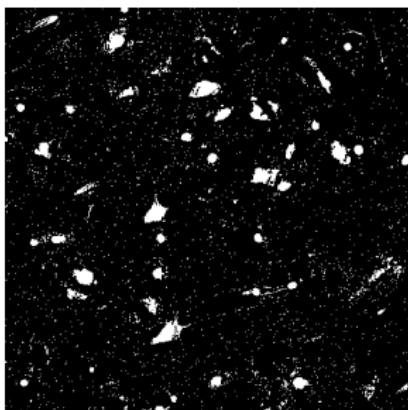
Dilatation

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



Ouverture

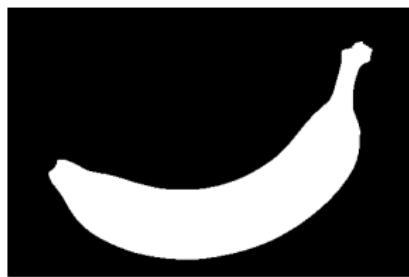
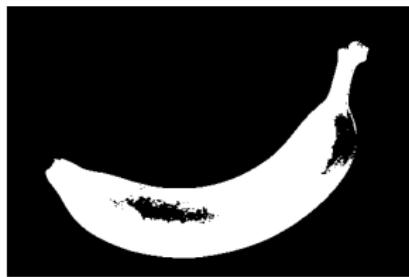
Ouverture : Érosion suivie d'une dilatation de l'image érodée



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

Fermeture

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

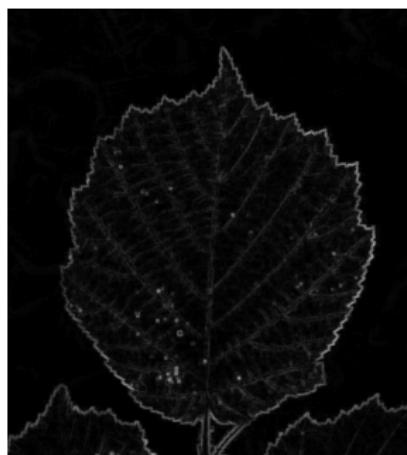
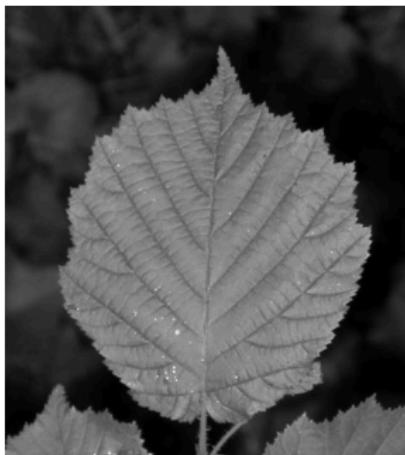


Intérêt : fusionne les structures proches, les trous sont comblés. Permet de recoller les composantes d'un objet.

Gradient morphologique

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

⇒ Donne des contours



Filtres (1/8)

Les filtres permettent de :

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

Filtres (2/8)

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

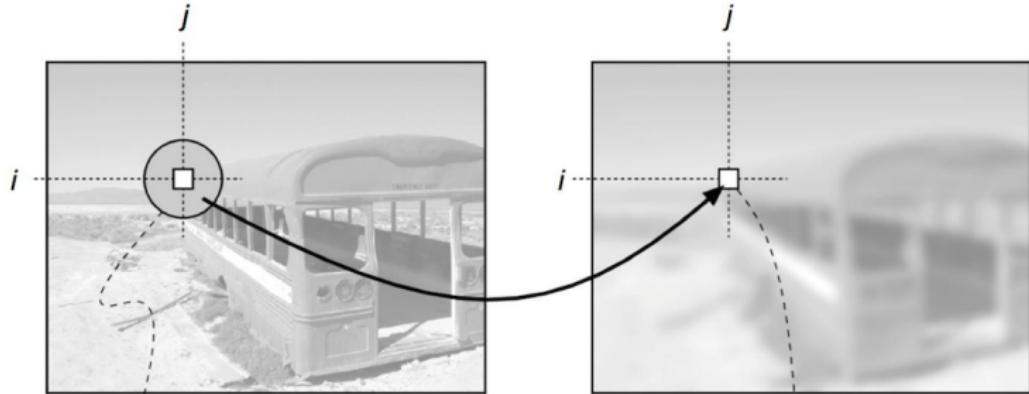
Importance de la **taille** du filtre (3x3, 5x5, 7x7...)

Filtres (3/8)

Exemple : moyenne dans un voisinage V de taille 3x3.

Nouvelle intensité d'un pixel en (i, j) :

$$I'(i, j) = \frac{1}{9} \sum_{(u, v) \in V(i, j)} I(u, v)$$



Filtres (4/8)

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

*

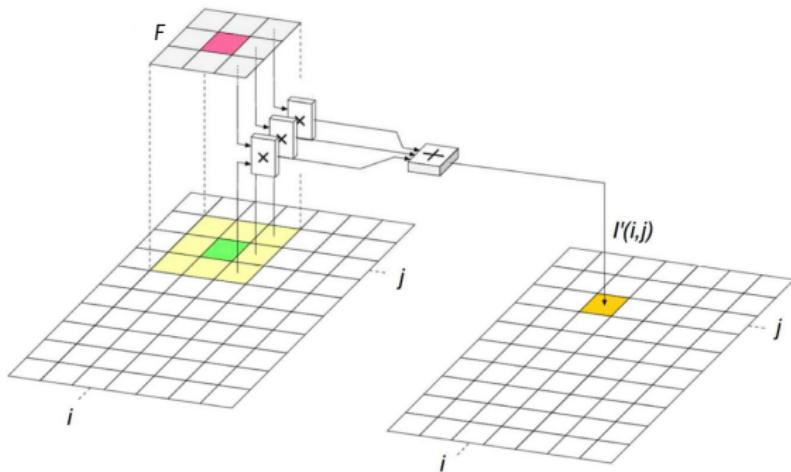
0	-1	0
-1	5	-1
0	-1	0

=

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 (5/8)

Produit de convolution : en images :



Filtres (6/8)

Algorithme :

1. f_t = filtre de taille t
2. Pour chaque pixel p de l'image :
 - 2.1 $I_{finale} = 0$
 - 2.2 Pour chaque pixel de coordonnées (x, y) dans la région de taille t :
 - $I_{finale} +=$ intensité du pixel en (x, y) multipliée par le coefficient en (x, y) du filtre f_t
 - 2.3 $I(p) = I_{finale}$

Filtres (7/8)

Gestion des bords : quel calcul lorsque le filtre est en dehors de l'image ?

Stratégies : (1) zero-padding, (2) étendre, (3) miroir, (4) périodique :



Filtres (8/8)

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
 3×3

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)

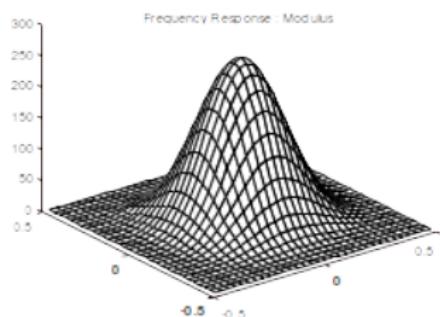
Inconvénients :

- 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 :

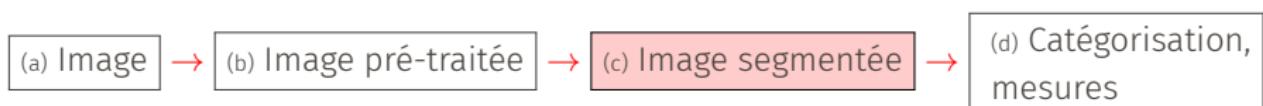
- Bons résultats bruit poivre et sel
- Élimine moins les détails de l'image que le filtre moyenneur

Inconvénients :

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

DÉTECTION DE CONTOURS

Position dans la chaîne de traitement



On s'intéresse ici à une autre méthode de détection d'objets : détection de contours.

Détection de contours (1/7)

Gradient morphologique : donne des contours sur des images binaires.



Détection de contours (2/7)

Idée : un contour est la **frontière entre deux régions d'intensité différentes**.

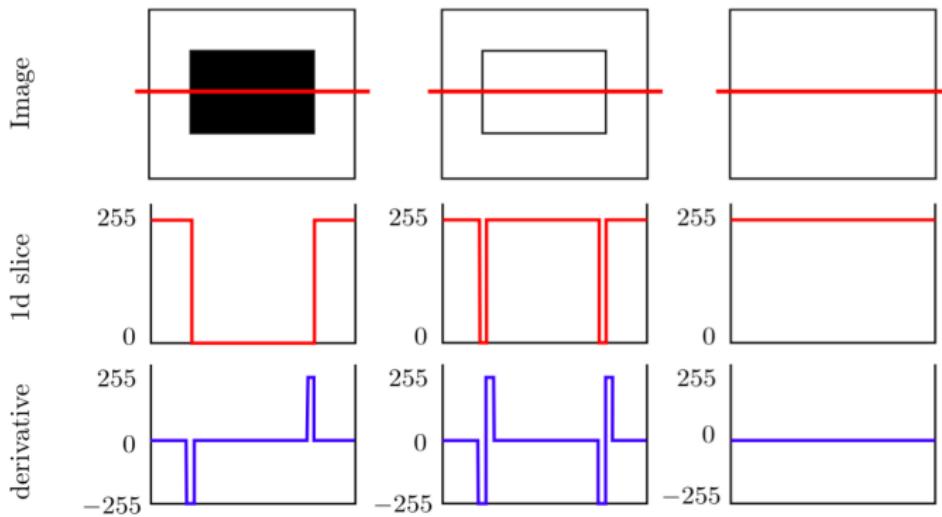
Principe : utiliser un filtre qui permette de détecter les pixels où se produisent les variations d'intensité.



Détection de contours (3/7)

Contour : bordure de l'objet

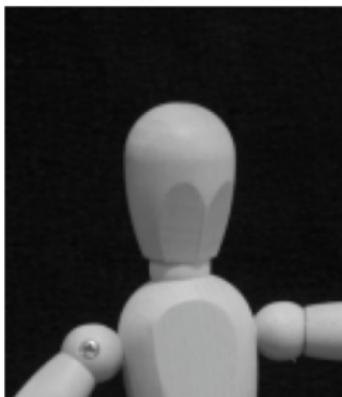
↳ Changements brusques en intensité → grande valeur de la dérivée



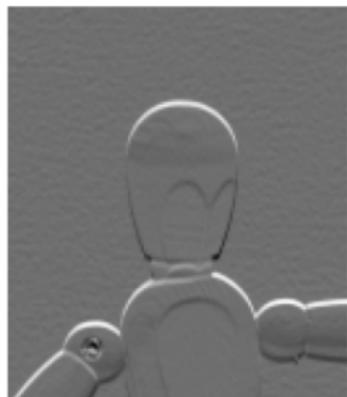
Détection de contours (4/7)

Différences finies : $f'(x) \approx f'(u) = \frac{f(u+1) - f(u-1)}{2}$

Gradient : Convolution avec un filtre $H = [-1, 0, 1]$



(a) Gradient ∇_h



(b) Gradient ∇_v

Détection de contours (5/7)

Filtres de Sobel : 3x3 :

1. Horizontal
2. Vertical

Donnent les **gradients**.

Les coefficients sont fixés pour donner **plus de poids** à la ligne/colonne courante.

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

(a) Filtre horizontal

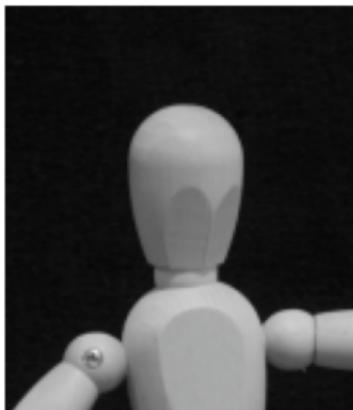
-1	0	+1
-2	0	+2
-1	0	+1

(b) Filtre vertical

Détection de contours (6/7)

L'image de contours finale est donnée par la **magnitude** des deux images de gradients :

$$I_{\text{contours}} = \sqrt{\nabla_h^2 + \nabla_v^2}$$



Détection de contours (7/7)

Résumé :



Filtres de
Sobel

Problèmes : Sensible au bruit, aux détails.

Algorithme de Canny (1/2)

Algorithme de détection de contours :

1. Appliquer un flou : généralement **filtre gaussien**
2. Calcul des gradients : **filtres de Sobel**
3. Amincissement du contour : **maxima locaux**
4. Seuillage : **deux seuils** sur la magnitude du gradient :
 - Si inférieur au seuil bas, **point supprimé**
 - Si supérieur au seuil haut, **point accepté**
 - Si entre seuils bas et haut, point accepté **si connecté** à un point accepté

Algorithme de Canny (2/2)

Comparaison Sobel vs. Canny :

