

Analyse d'image: Segmentation et Reconnaissance de Formes

Contenu du Chapitre

I. Analyse par Segmentation d'images

- Segmentation d'Image
- Approches de segmentation
- Segmentation par Région
- Méthodes globales : Segmentation par Seuillage d'Histogramme
 - Segmentation par Binarisation
 - Seuillage multiniveaux
- Méthodes locales : Segmentation par transformation de régions
- Segmentation par Croissance de Région (Region Growing)
- Segmentation par Partage de Région (region splitting)
- Split-and-merge

II. Traitement haut-niveau

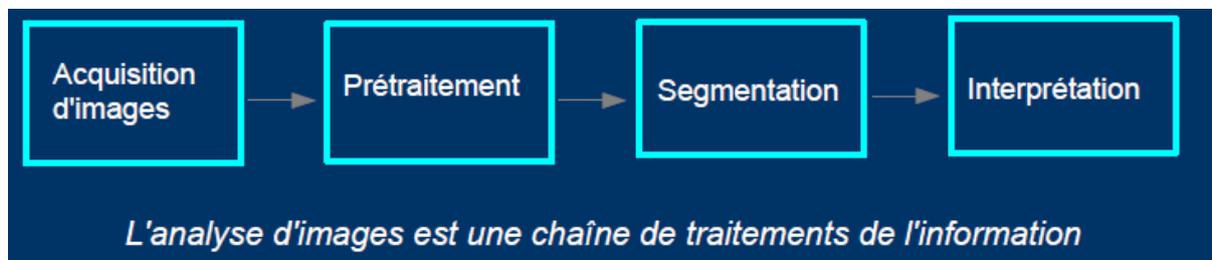
Généralités : analyse d'images et reconnaissance de formes

- Introduction
- Reconnaissance de formes (*Pattern recognition*)
- Méthodes de reconnaissance et Applications
- Processus d'analyse pour une RDF
 - Extraction de Caractéristiques
 - Reconnaissance de formes par classification de caractéristiques
- Classification de données pour la reconnaissance de formes
 - Choix des caractéristiques
 - Choix du classifieur
 - Distance minimale
 - Machine à vecteurs de support (*SVM : support vector machine*)
- Conclusion

I. Analyse par Segmentation d'images

L'analyse des images regroupe plusieurs disciplines que l'on classe en deux catégories :

- Les processus de *bas-niveaux*, qui nécessitent très peu d'informations sur le contenu des images. Il s'agit des processus de filtrage, d'amélioration et de restauration d'images, processus que l'on regroupe sous le terme de traitement d'images, ainsi que l'extraction d'indices.
- Les processus de *haut-niveaux*, qui fonctionnent en aval de deux de bas niveaux, et qui peuvent nécessiter des informations sur le contenu des images. Il s'agit de la reconnaissance de formes, de la reconstruction 3-D, les processus cognitifs de façon générale.



Cette fonction est déterminante car son résultat conditionne le traitement appliqué au produit contrôlé (pour la suite de la fabrication ou mise au rebut). Le système assurant cette opération doit tout d'abord satisfaire aux critères devenus incontournables dans l'industrie comme partout : puissance, flexibilité, simplicité de mise en oeuvre et convivialité d'emploi. Le système de vision doit aussi, et surtout, répondre à un nouvel impératif : la **robustesse**. L'analyse de l'image repose en effet sur des outils permettant la reconnaissance de forme de l'objet à contrôler. Pour être efficaces, "robustes", ces outils doivent pouvoir reconnaître une forme tout comme le ferait l'oeil humain, c'est-à-dire en s'affranchissant des variations et en étant capables de distinguer entre ce qui est acceptable et ce qui ne l'est pas.

Voici les règles auxquelles doivent se conformer ces outils en général:

- trouver les objets malgré les changements de taille et d'orientation (0-360°)
- maintenir la précision malgré les changements de taille et d'orientation
- maintenir la vitesse malgré les changements de taille et d'orientation
- tolérer les variations d'éclairage et les ombres
- tolérer les manques et les inclusions, les objets se touchant ou se recouvrant

- s'affranchir du flou, du manque de contraste et du bruit de l'image
- accepteront les modèles aux contours peu marqués
- fonctionner sur des plates-formes standard

➤ *Communication avec l'environnement*

Le résultat de l'analyse ("bon ou mauvais") doit être transmis aux automatismes qui prennent en charge la suite des opérations, ainsi qu'aux opérateurs qui surveillent le processus de fabrication. Le système de vision doit donc être équipé :

- d'E/S TOR pour gérer les signaux de base (présence pièce, signal d'éjection des défauts),
- d'un écran opérateur avec une interface adaptée et configurable,
- et, pour répondre à une demande plus récente mais en forte croissance, d'une interface avec un réseau standard de communication.

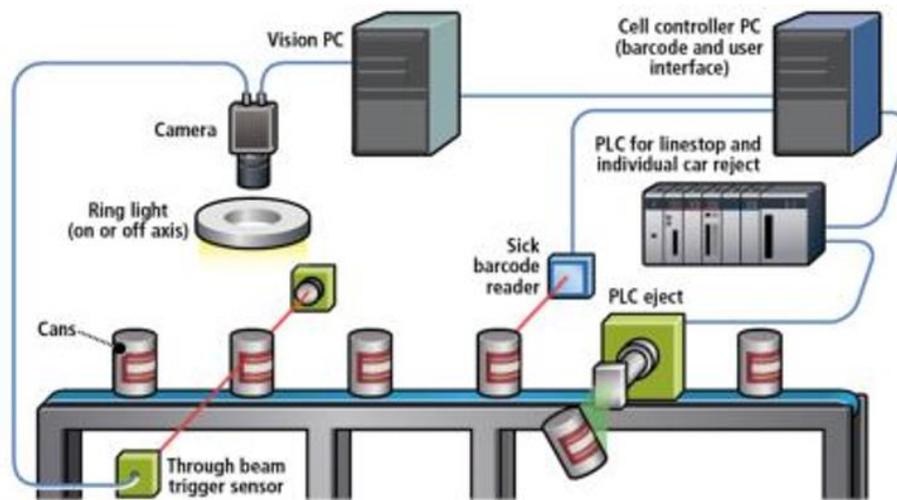


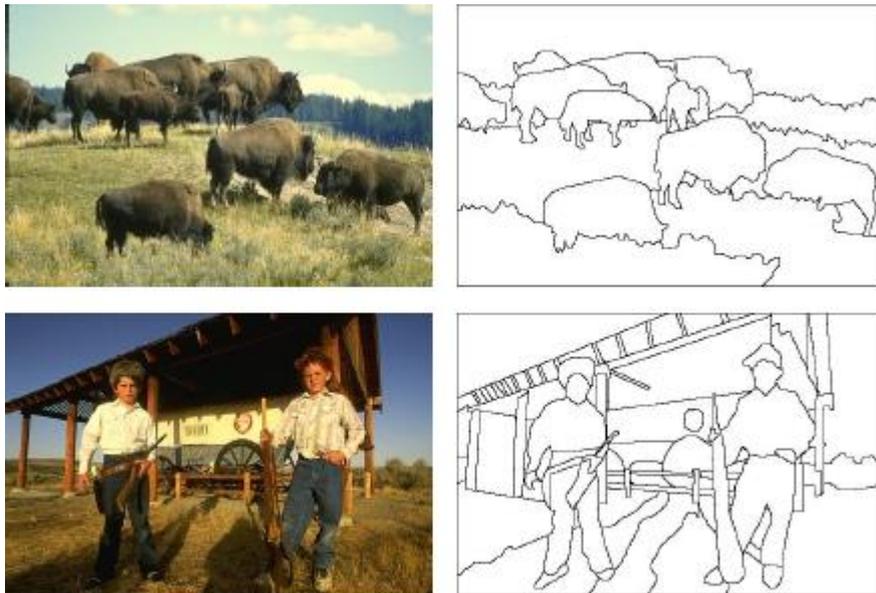
Schéma d'une Installation d'Inspection par Système de Vision Industrielle

• **Segmentation d'Image**

En vision par ordinateur, la segmentation d'image est le processus de partitionner une image numérique en plusieurs segments ou régions homogènes (ensembles de pixels, également appelés super-pixels). Le but de la segmentation est de simplifier et / ou changer la représentation d'une image en quelque chose de plus significatif et plus facile à analyser. La segmentation est généralement utilisée pour localiser des objets et des limites (lignes, courbes, etc.) dans les images. Plus précisément, la segmentation d'image est le processus d'attribution d'une étiquette à chaque pixel

d'une image telle que les pixels avec la même étiquette partagent certaines caractéristiques. Segmenter une image signifie donc la diviser en "régions homogènes", selon un ou plusieurs attributs donnés (niveau de gris, texture, couleur, etc). On appelle "bords" ou "contours" de l'image les frontières entre ces régions. La notion de bonne segmentation/bonne détection de contours dépend fortement du type d'image à traiter et des applications envisagées.

Quelques segmentations proposées par des humains, disponibles sur le site : <http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/segbench/>



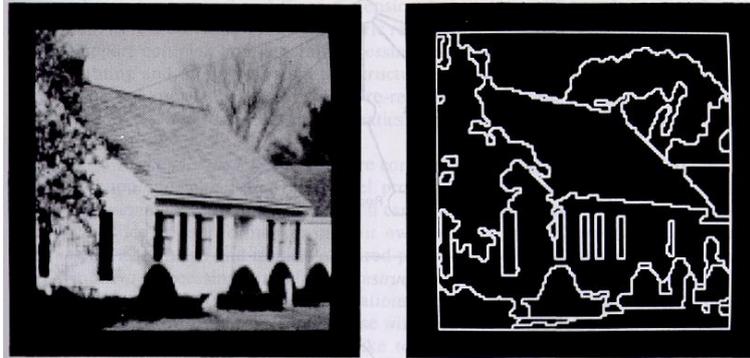
Exemple de Segmentation d'image effectuée par des humains

➤ Buts et Difficultés de la Segmentation

- Le but de la segmentation est de partitionner une image en régions (par exemple, séparer les objets de l'arrière-plan)



- Les résultats de la segmentation sont très importants pour déterminer le succès ou échec éventuel de l'analyse d'image

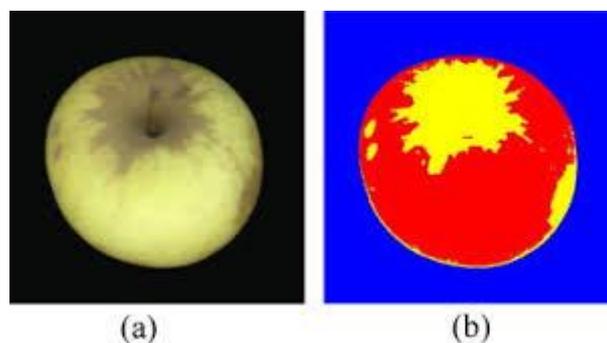


- La segmentation est un problème très difficile en général

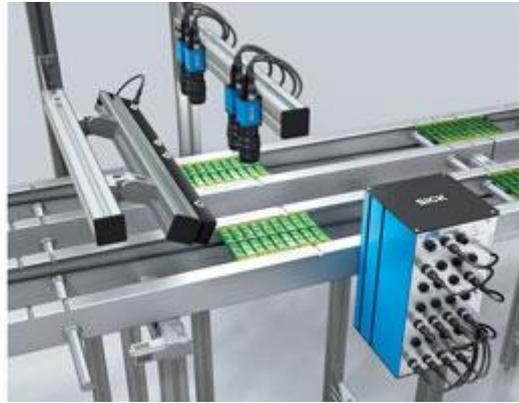


Augmentation de la précision et de la robustesse

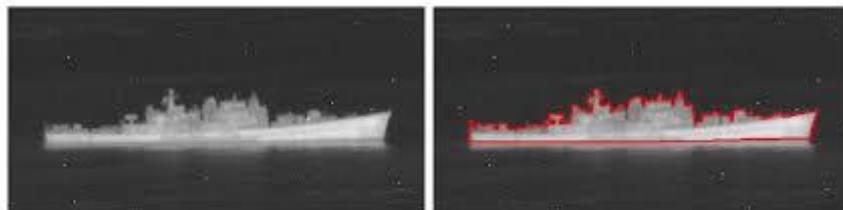
- Introduire suffisamment de connaissances sur le domaine d'application



- Contrôler l'environnement (par exemple, dans les applications industrielles)



- Sélectionnez le type de capteurs pour améliorer les objets d'intérêt (par exemple, utiliser l'imagerie infrarouge pour les applications de reconnaissance de cible)



• Approches de segmentation

Les deux approches utilisées en segmentation d'image sont principalement l'approche région qui est basée sur la notion d'homogénéité d'une région et l'approche contour qui suppose que toute région est délimitée par un contour fermé ou bord.

❖ Les approches basées sur les bords:

- a- Utilisent les limites des régions pour segmenter l'image
- b- Détectent les changements brusques d'intensité (discontinuités)
 - Méthodes de détection et suivi de contours : Avant de pouvoir extraire des lignes dans l'image, il faut d'abord avoir une image de contours. Ces contours s'obtiennent facilement à partir de l'image binaire ou de niveaux de gris. Plusieurs opérateurs sont possibles pour avoir les contours d'une image : on peut faire une convolution avec un filtre de Sobel, ou de Prewitt ou encore utiliser l'algorithme de Canny.
 - Une fois que l'image des contours est obtenue, un algorithme d'extraction de lignes ou droites et de fermeture de contours peut être appliqué. On peut citer par exemple l'algorithme classique de la transformée de Hough pour représenter les droites et certaines courbes telles que les cercles et ellipses de façon paramétrique.

❖ Les approches basées sur les régions:

- Utilisent la similitude entre les pixels d'une même région et la disparité avec les autres pixels pour trouver différentes régions selon certains critères d'homogénéité, *couleur, texture, niveau de gris,...*

Théoriquement, les deux approches devraient donner des résultats identiques mais cela n'est pas vrai dans la pratique

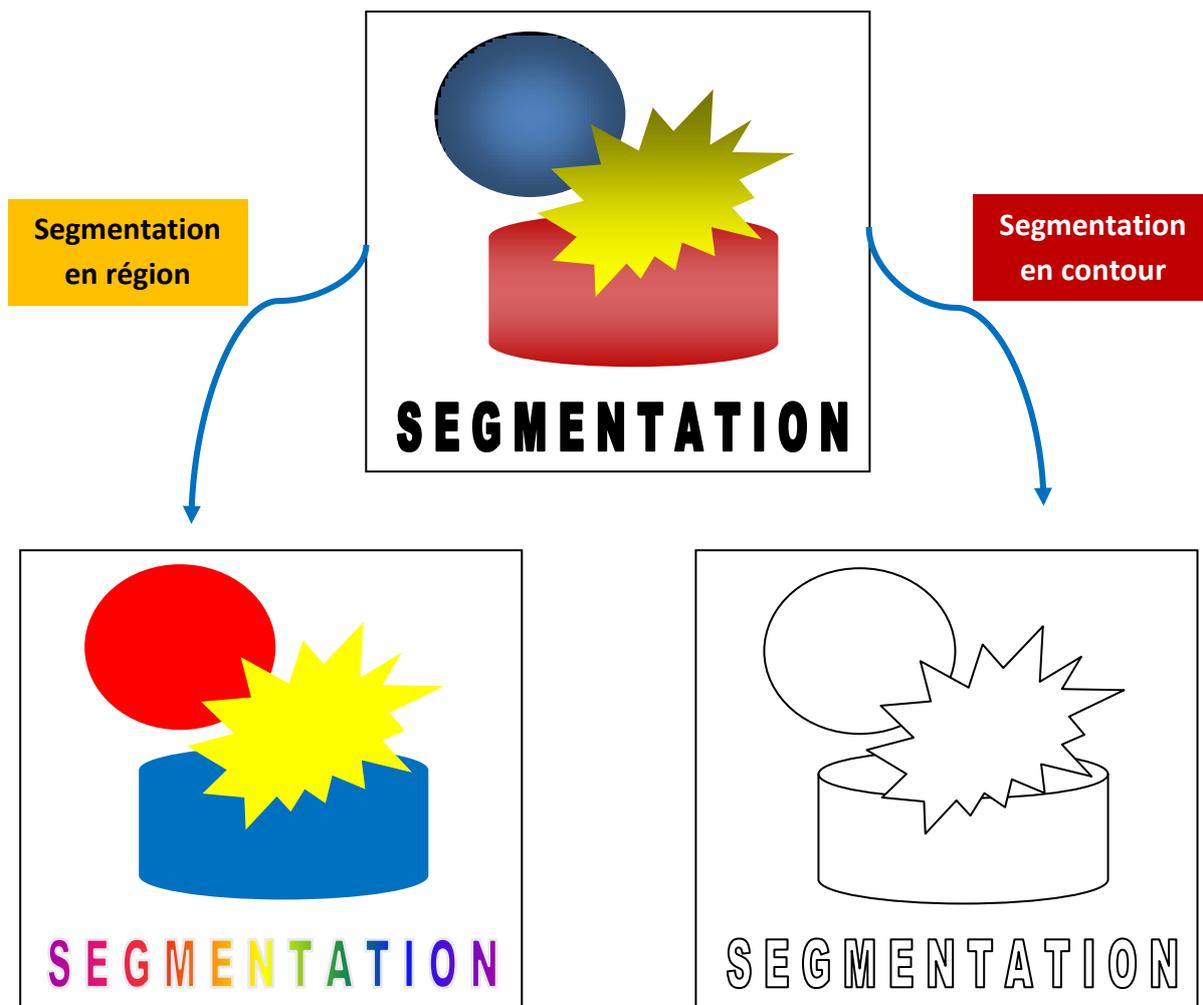
Deux Principales Approches de la Segmentation

Approche par Région :

Recherche des régions de même intensité, couleur ou texture etc...

Approche par Contour :

Recherche des contours séparant les différentes régions



Dans cette partie de ce chapitre, l'approche de segmentation en région sera plus ou moins exposée en détail. Par contre, l'approche contour a été développée dans un chapitre précédent.

• Segmentation par Région

• Définitions et Rappels

Une procédure composée d'un seuillage suivi d'une décomposition en composantes connexes est une segmentation **statique**. Elle consiste à sélectionner les points des zones d'intérêt de l'image, puis à les regrouper en **taches** ou **composantes connexes**.

Les informations numériques extraites sont de nature :

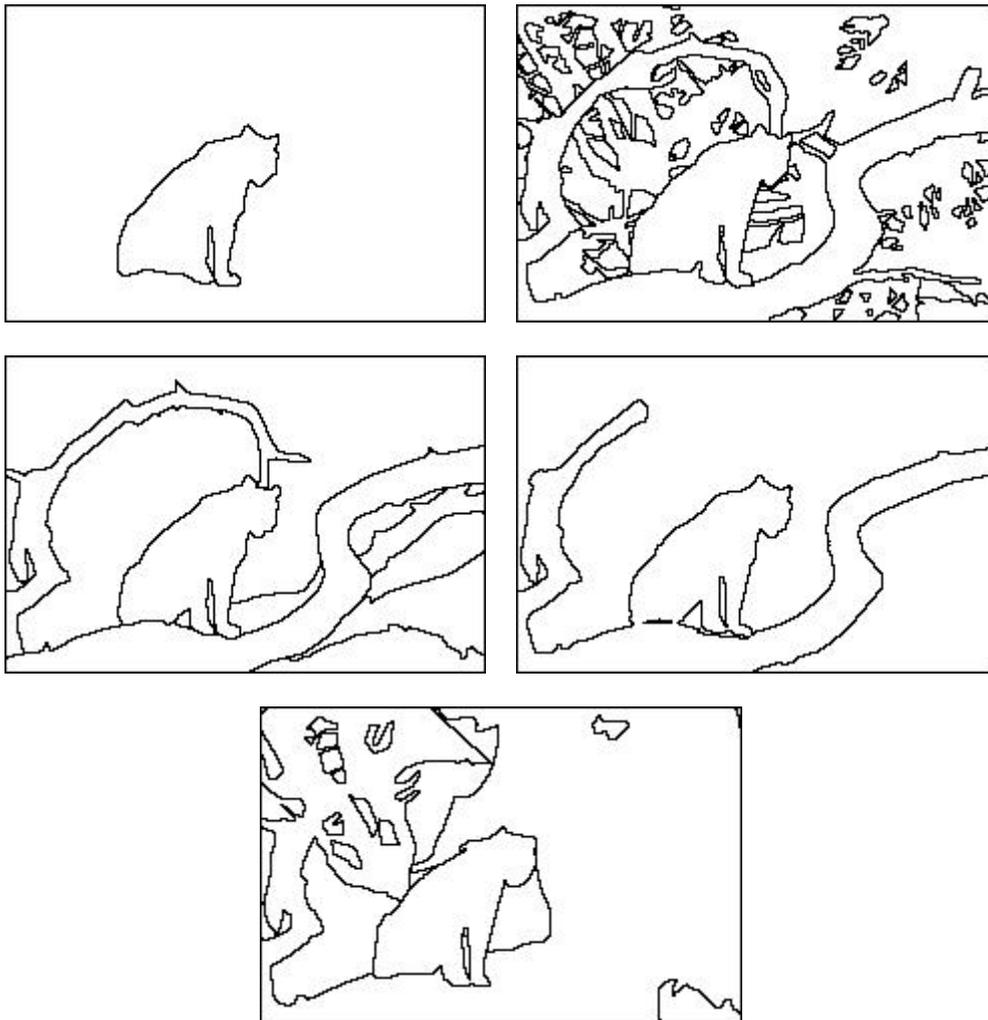
- géométrique : permettant la localisation dans l'image :
 - surface,
 - centre de gravité,
 - boîte englobante ;
- photométrique : permettant l'identification :
 - niveau, ou couleur.

Composante connexe : Une composante connexe (*connected component*) est un groupe de pixels tel qu'on puisse aller d'un pixel de ce groupe à un autre pixel de ce groupe en passant par des pixels du même groupe voisins entre eux.

Remarques

- Chaque région R_i est une composante connexe.
- Le résultat de la segmentation n'est pas unique (dépend du critère d'homogénéité, de la méthode, de l'initialisation, etc.) .





Segmentation ?

- ✓ Partition d'une image en un ensemble de **régions**

⇒ **Région ?**

- ✓ **Zone homogène** dans l'image :

- niveau de gris,
- couleur,
- texture,
- gradient,
- ...

⇒ **Principes de base de la segmentation?**

- ✓ **Similarité**
- ✓ **Proximité spatiale**

Formulation de la segmentation :

- ✓ Partition en régions $\{R_i\}$ de l'image I :
- tout pixel appartient à une région :

$$\bigcup_i R_i = I$$
- aucun pixel n'appartient à plus d'une région :

$$\forall i, j (R_i \cap R_j) = \emptyset$$
- proximité (cohérence) spatiale :

$$\forall i, \forall p, p' \in R_i \rightarrow p \text{ est connecté à } p'$$
- cohérence de caractéristique de région, pour un critère d'homogénéité P :

$$\forall i, P(R_i) = \text{vrai}$$

pour R_i et R_j adjacents et $i \neq j, P(R_i \cup R_j) = \text{faux}$

⇒ Définition

- Segmenter par régions est équivalent à partitionner une image en un sous-ensemble de zones connexes.

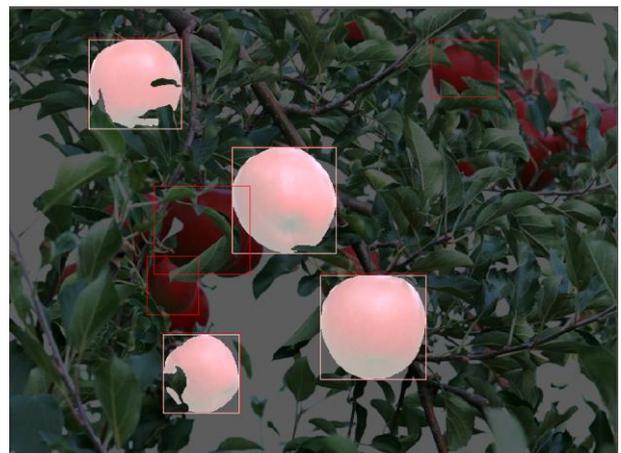
⇒ Contexte

- La segmentation repose sur une recherche de zones connexes de l'image possédant des attributs communs (luminosité, texture, ...)

⇒ Intérêt

- L'intérêt est de pouvoir ensuite manipuler les zones connexes détectées de façon à en extraire différentes caractéristiques : forme, position, taille, ...

un exemple d'utilisation de segmentation
par régions : étiquetage de pièces,
d'objets,



• Méthodes de Segmentation : globale ou locale ?

Dans l'approche de segmentation en régions on utilise souvent soit des méthodes globales, par exemple segmentation par seuillage d'histogramme, soit des méthodes locales de partition profitant des propriétés de voisinage des pixels.

Les méthodes par histogrammes sont en général :

- Rapides à calculer
- peu sensibles au bruit, mais elles n'intègrent pas (ou peu) d'information géométrique ou topologique sur les régions.

Ce sont des méthodes globales, au sens où la décision d'appartenance d'un pixel à une région dépend toujours de l'image entière. A l'autre extrême, on peut considérer les méthodes locales, pour lesquelles l'étiquetage d'un pixel dépend uniquement, ou essentiellement, de sa relation avec son voisinage. L'exemple caractéristique est la segmentation par croissance de région (region growing). Ces algorithmes intègrent naturellement les propriétés topologiques, mais aussi parfois géométriques des régions.

• Méthodes globales

❖ Segmentation par Seuillage d'Histogramme

• Principe

Une segmentation de l'histogramme de niveaux de gris h_I donne une segmentation de l'image I en classes selon le (ou les) seuils choisis (s). Le seuillage a pour objectif de segmenter une image en plusieurs classes en n'utilisant que l'histogramme. On suppose donc que l'information associée à l'image permet à elle seule la segmentation, i.e. qu'une classe est caractérisée par sa distribution de niveaux de gris. A chaque pic de l'histogramme est associée une classe. Il existe de très nombreuses méthodes de seuillage d'un histogramme. La plupart de ces méthodes s'appliquent correctement si l'histogramme contient réellement des pics séparés. De plus, ces méthodes ont très souvent été développées pour traiter le cas particulier de la segmentation en deux classes

(i.e. passage à une image binaire) et leur généralité face aux cas multi-classes n'est que très rarement garantie.

- **Segmentation par Binarisation**

Elle consiste à trouver un seuil μ_{seuil} de niveau de gris optimal. L'image d'entrée est une image d'intensité $I(p)$ du pixel $p = (x, y)$ en niveaux de gris et l'image résultante sera une image binaire $B(p)$ en noir et blanc. Ce traitement nécessite le réglage d'un paramètre appelé le seuil, noté μ_{seuil} . Si la valeur du pixel P de l'image dépasse le seuil fixé alors la valeur résultante du pixel est égale à 1 qui correspond à un point objet (*foreground pixel*) sinon il prend la valeur 0 ce qui correspond à un point du fond de l'image (*background pixel*).

$$B(p) = \begin{cases} 1 & \text{si } I(p) \geq \mu_{seuil} \\ 0 & \text{autrement} \end{cases}$$

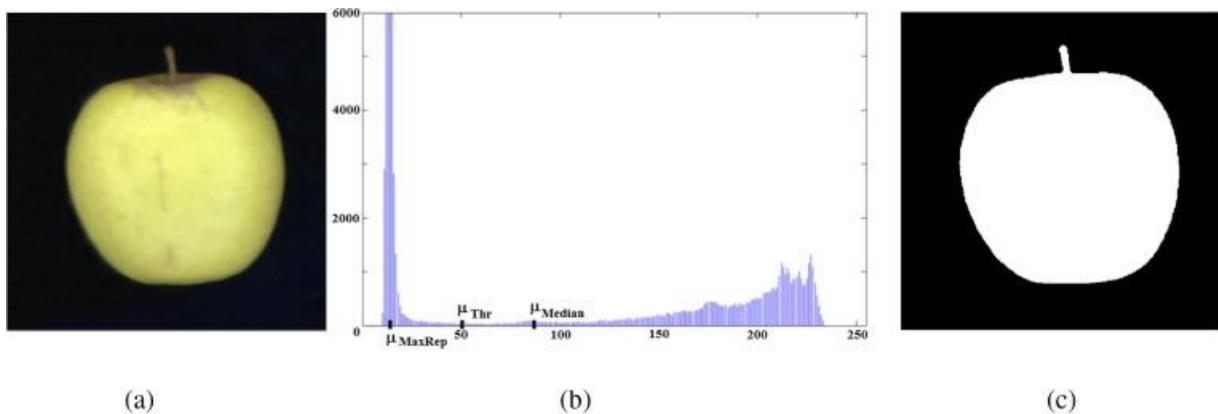
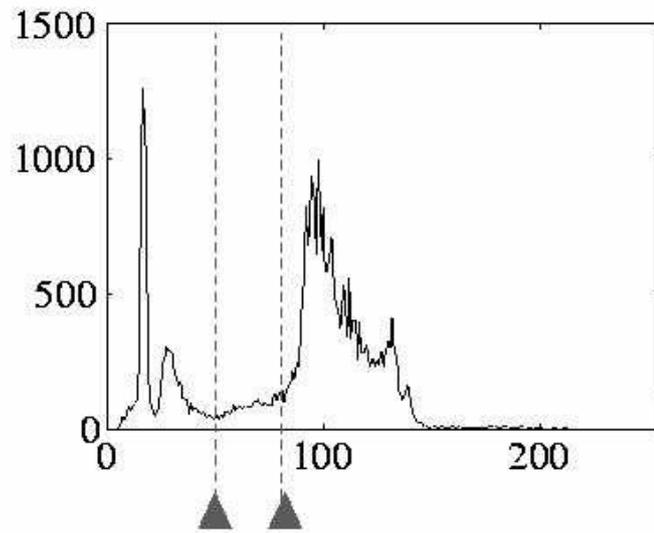


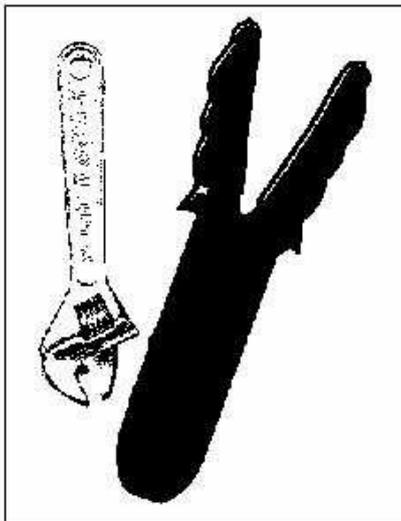
Image Originale



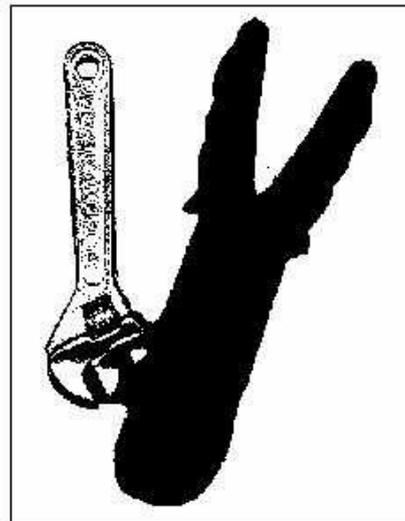
Histogramme



➤ La binarisation dépend du choix du seuil : 50 75



Seuil = 50



Seuil = 75

- **Seuillage multiniveaux**

revient à trouver n seuils optimaux $\mu_1 < \mu_2 < \dots < \mu_n$ dans h , i.e. à segmenter h en $(n + 1)$ parties distinctes R_i , $i = 1, \dots, n$. L'image segmentée est alors : $I_{segmentée} = \cup_i R_i$ avec $I(R_i) = moyenne(I(p))$ où $\mu_i < I(p) < \mu_{i+1}$

Ces méthodes supposent n connu ou pas, méthodes statistiques (mélange de gaussiennes.), etc.

Ces méthodes sont globales : elles n'utilisent pas l'information spatiale sur les pixels.

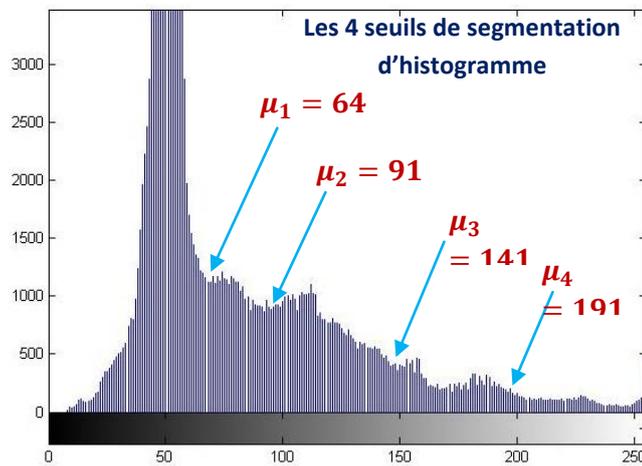


Image couleur

RGB2gray
→



Image en niveaux de gris



Histogramme

Histogramme de l'image en niveaux de gris

Segmentation multi-seuils

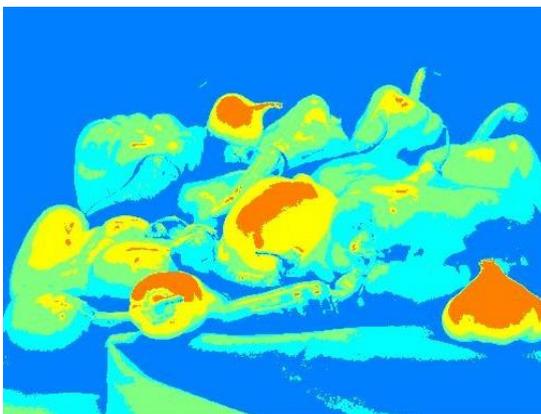


Image segmentée en 5 régions colorées

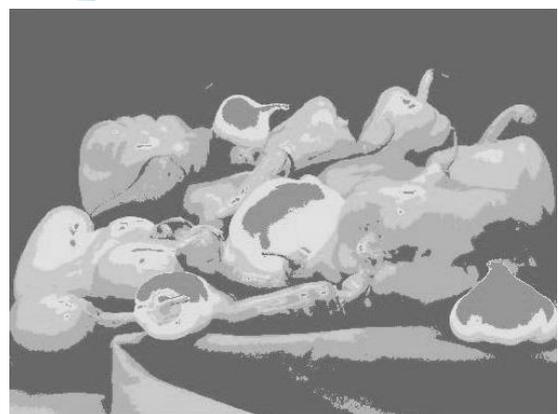


Image segmentée en 5 régions de gris

- **Seuillage local**

D'autres méthodes font un seuillage local, on calcule un seuil optimal $\mu(I(p))$ à partir des caractéristiques locales de l'image $I(p)$.

- **Méthodes locales**

- **Segmentation par transformation de régions**

Le but de la **Segmentation en Régions** va être de regrouper entre eux les pixels d'une image vis à vis d'un critère d'homogénéité H et qui peut s'exprimer de la manière suivante :

La segmentation d'une image I par rapport au critère H est une partition de l'image I en régions homogènes R_1, \dots, R_n telles que :

- $\forall i, j (R_i \cap R_j) \neq \emptyset$
- $\forall i, R_i \neq \emptyset$
- $\cup_i R_i = I$

⇒ **Approche fondée sur un partitionnement**

- Le traitement consiste à créer une partition de l'image I en **sous-ensembles de pixels connexes R_i appelés régions**



Les algorithmes de segmentation en régions sont généralement composés de plusieurs étapes successives. En effet, il est très difficile, voire quasiment impossible, de regrouper correctement les pixels pour former des régions en une seule procédure, car les régions sont généralement homogènes au centre et le sont de moins en moins plus on s'en éloigne. L'idée est de partir des zones homogènes puis de relâcher progressivement les contraintes. Ceci nous amène à regrouper des pixels entre eux en un premier temps, puis des régions ensuite, ce qui ne peut pas se faire avec le même algorithme.

⇒ *Différentes Méthodes*

- **Partage de régions**
- **Croissance de régions**
- **Réunion de régions**

La segmentation initiale peut être prise de deux manières différentes, ainsi qu'à partir d'une combinaison de ces deux manières :

- **Méthode de Partage de régions** : à partir de l'image entière, et en la découpant récursivement jusqu'à obtenir des régions initiales suffisamment homogènes (algorithme de **Découpage** ou **Split**)
- **Méthode de croissance de Régions** : à partir des pixels de l'image, que l'on regroupe ou fusionne les uns aux autres (algorithme de **Fusion** ou **Merge**)
- **Méthode mixte** : à partir d'un regroupement quelconque, en utilisant successivement le découpage, puis la fusion. Une telle procédure est intéressante, lorsque l'on dispose d'une segmentation proche, ce qui est le cas lorsque l'on segmente une séquence d'images, où l'on dispose des résultats de la segmentation de l'image précédente pour segmenter l'image courante.

⇒ *Contexte*

- Le problème de segmentation par régions est mal posé car il n'existe pas de segmentation idéale
- Le choix d'une méthode est lié :
 - ✓ à la nature des images (éclairage, texture, ...)
 - ✓ à la forme des primitives que l'on veut extraire ultérieurement des régions détectées
 - ✓ aux éventuelles contraintes de temps

⇒ *Notion de prédicat d'homogénéité*

- ✓ Un prédicat est une proposition logique dont la valeur dépend de son argument
- ✓ R_k est homogène si et seulement si $P(R_k)$ est vraie

avec :

- I , une image,
- R_k une région quelconque de I ,
- P , un prédicat définissant l'homogénéité d'une région

Exemple : $P(R_k) \Leftrightarrow (\text{variance des intensités des pixels de } R_k < \text{seuil})$

✓ Méthodes qui s'appuient sur les notions de prédicat d'homogénéité et de partition en régions $\{R_i\}$ de l'image I vérifient que:

- $\forall i, j (R_i \cap R_j) \neq \emptyset$
- $\forall i, R_i \neq \emptyset$
- $\cup_i R_i = I$

Quelques critères d'homogénéité d'une région

→ *Contraste sur la région*

$$P(R_k) = \text{vrai} \Leftrightarrow (\text{Max}_{R_k} \{ I(x,y) \} - \text{Min}_{R_k} \{ I(x,y) \}) < \sigma$$

avec :

- I , une image,
- R_k une région quelconque de I ,
- σ , un seuil

→ *Ecart-type sur la région*

$$P(R_k) = \text{vrai} \Leftrightarrow \sqrt{\frac{1}{N} \sum_{R_k} (I(x,y) - m)^2} < \sigma$$

avec :

- I , une image,
- R_k une région quelconque de I ,
- σ , un seuil
- $N = \text{Card}(R_k)$
- $m = \frac{1}{N} \sum_{R_k} I(x,y)$

→ *Entropie (mesure de la quantité d'information)*

$$P(R_k) = \text{vrai} \Leftrightarrow -\sum_{R_k} (p(I(x,y)) \log(p(I(x,y)))) < \sigma$$

avec :

- I , une image,
- R_k une région quelconque de I ,
- σ , un seuil
- $p(I(x,y))$, la probabilité d'apparition du niveau de gris $I(x,y)$

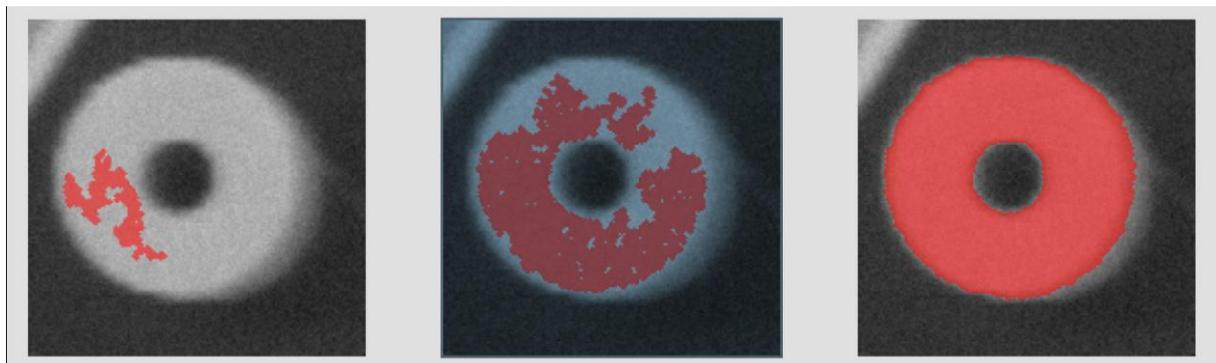
- **Segmentation par Croissance de Région** (Region Growing)

⇒ **Définition grossière**

Méthode fondée sur l'agglomération de pixels voisins à une région en maintenant vrai le prédicat d'homogénéité choisi

⇒ **Principe général**

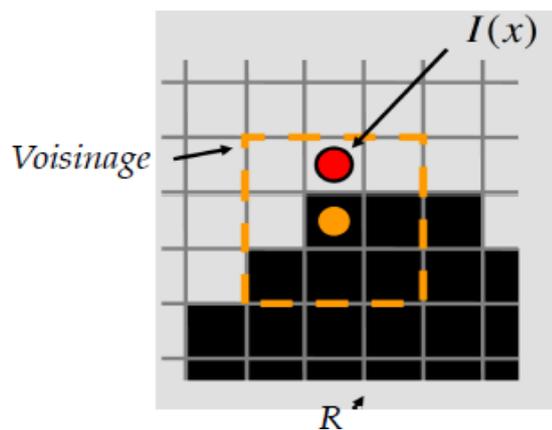
- Approche "ascendante"
- Le processus commence à partir d'un pixel (ou d'un groupe de pixels connexes) d'amorce (germe)
- Considération des pixels voisins et analyse du prédicat d'homogénéité
- Croissance de la région jusqu'au critère d'arrêt (plus aucun pixel ne peut être ajouté sans compromettre l'homogénéité de la région)



⇒ **Mise en oeuvre avec un exemple de prédicat d'homogénéité**

- Homogénéité suivant la moyenne de la région : μ_R
- μ_R est mise à jour à chaque itération,
- Pour chaque nouveau pixel voisin $I(x)$, le prédicat à vérifier est :

$$|I(x) - \mu_R| < \sigma$$



avec :

- I , une image,
- R , une région quelconque de I à détecter,
- σ , un seuil
- μ_R la moyenne de la région en croissance

▪ Choix des germes

⇒ Propositions

- Germes sont donnés par l'utilisateur (cadre supervisé)
- Germes tirés au hasard parmi les pixels qui n'ont pas été traités
- Germes pris au fur et à mesure des besoins, dans le sens du balayage
- Germes sélectionnés par analyse de l'histogramme

▪ Avantages et limites

⇒ Avantages

- ✓ Implémentation de cette méthode relativement simple
- ✓ Assez rapide

⇒ Limites

- ✓ Algorithmes très sensibles au bruit,
- ✓ Obtention de frontières non-régulières
- ✓ Peu efficace en cas de dégradés d'attributs
- ✓ Influence des nombre et ordre de prise en compte des germes, des voisins
- ✓ Choix critique des attributs et des valeurs des seuils
 - Risque de sur-segmentation ou de sous-segmentation
 - Dépend du type d'images à traiter

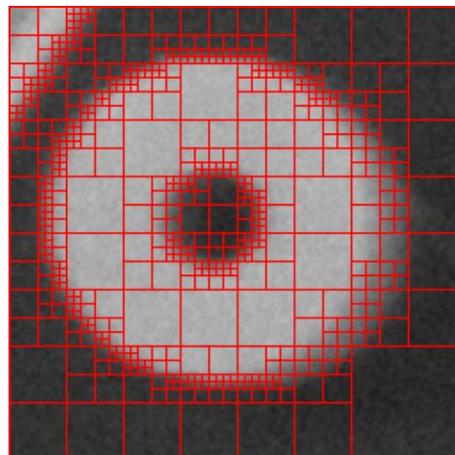
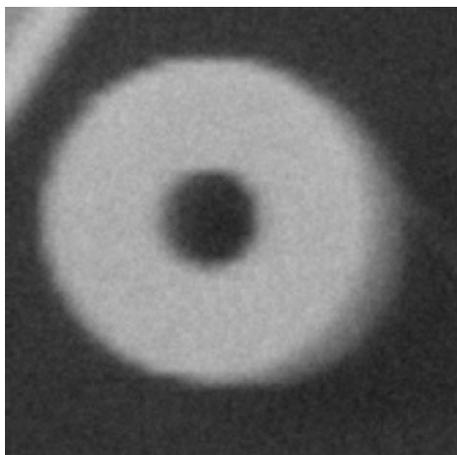
- **Segmentation par Partage de Région** (region splitting)

⇒ **Définition**

L'objectif de cette méthode est de partager l'image à traiter de façon automatique en un ensemble de régions homogènes au regard d'un prédicat d'hétérogénéité P' choisi.

⇒ **Principe général**

- Approche "descendante"
- En général, au départ le prédicat d'homogénéité P n'est pas respecté sur l'image entière, c'est-à-dire que P' est vérifié ($P' = \neg P$)
- Si P' est vérifié, on divise en régions
- Arrêt du processus lorsqu'aucune région n'est hétérogène



Quelques critères d'hétérogénéité d'une région (duale de l'homogénéité)

→ *Contraste sur la région*

$$P'(R_k) = \text{vrai} \Leftrightarrow (\text{Max}_{R_k} \{ I(x,y) \} - \text{Min}_{R_k} \{ I(x,y) \}) > \sigma$$

avec :

- I , une image,
- R_k , une région quelconque de I ,
- σ , un seuil

→ *Ecart-type sur la région*

$$P'(R_k) = \text{vrai} \Leftrightarrow \sqrt{\frac{1}{N} \sum_{R_k} (I(x,y) - m)^2} > \sigma$$

avec :

- I , une image,
- R_k , une région quelconque de I ,
- σ , un seuil
- $N = \text{Card}(R_k)$
- $m = \frac{1}{N} \sum_{R_k} I(x,y)$

→ *Entropie (mesure de la quantité d'information)*

$$P'(R_k) = \text{vrai} \Leftrightarrow -\sum_{R_k} (p(I(x,y)) \log(p(I(x,y)))) > \sigma$$

avec :

- I , une image,
- R_k , une région quelconque de I ,
- σ , un seuil
- $p(I(x,y))$, la probabilité d'apparition du niveau de gris $I(x,y)$

Split-and-merge : L'algorithme Split_and_merge décompose le processus de segmentation en deux étapes. L'image initiale peut être une première partition résultant d'une analyse grossière ou bien l'image brute. Dans la première étape, ou division (**split**), on analyse individuellement chaque région X_i . Si celle-ci ne vérifie pas le critère d'homogénéité, alors on divise cette région en blocs (le plus généralement en 4 quadrants) et l'on réitère le processus sur chaque sous-région prise individuellement. On peut tout à fait initier le processus en considérant que la première région est composée de toute l'image. Dans la deuxième étape, ou réunion (**merge**), on étudie tous les couples de régions voisines (X_k, X_i). Si l'union de ces deux régions vérifie le critère d'homogénéité, alors, on fusionne les régions. La principale difficulté de cette approche réside bien sûr dans le parcours de l'ensemble de tous les couples de régions voisines.

II. Traitement haut-niveau

Généralités : analyse d'images et reconnaissance des formes

Analyse d'Images  Interpretation du contenu

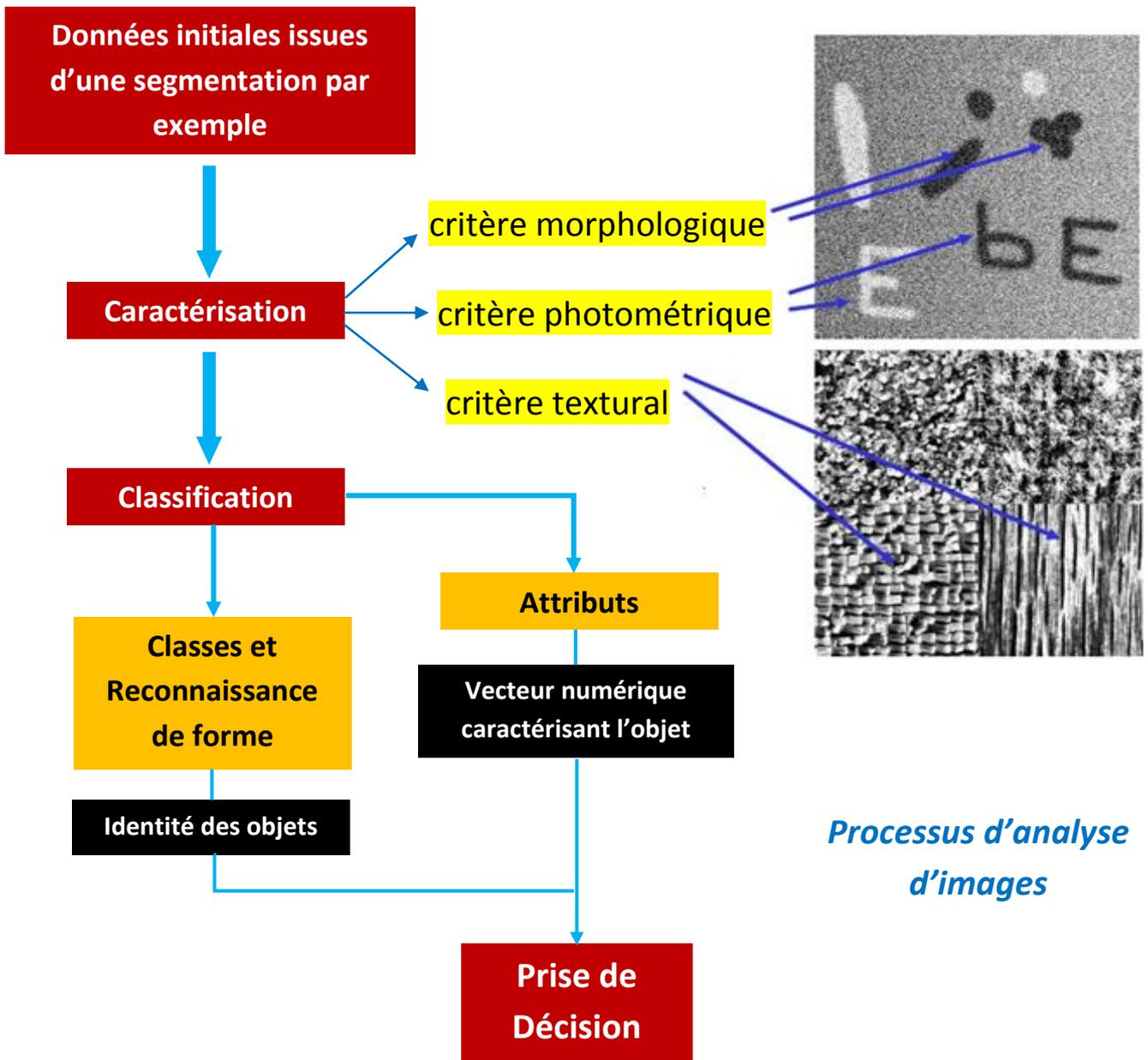
- **Segmentation d'images** : décomposition d'une image en régions homogènes (selon un critère donné).
- **Détection d'objets d'intérêt** : localiser (et extraire) des zones caractéristiques.
- **Reconnaissance d'objets d'intérêt** : identifier un objet – lui assigner une classe d'appartenance.

• Introduction

En analyse d'images, et plus particulièrement la segmentation et la détection d'objets d'intérêt, on va tenter d'associer à chaque pixel (ou ensemble de pixels) un label (classification) en s'appuyant sur un certain type d'information :

- Niveau de gris ou couleur : critère photométrique
- Distribution spatial du voisinage : critère textural
- Forme géométrique : critère morphologique





• Reconnaissance de formes (*Pattern recognition*)

La reconnaissance des formes (RDF) consiste à identifier ou classer des « formes » ou objets en se basant sur certaines de leurs caractéristiques.

- Concevoir des systèmes automatiques ou semi-automatiques qui reconnaissent les formes qu'on leur présente.

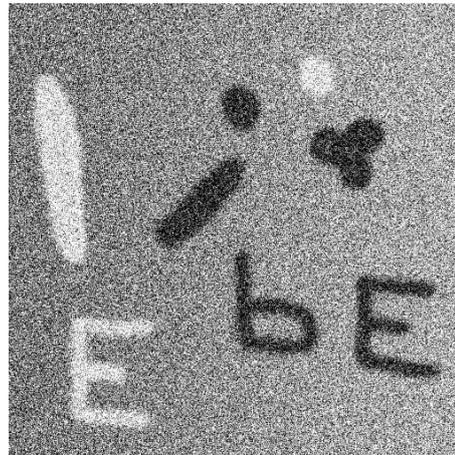
Principe : identifier des formes dans des images. Le mot 'forme' est à prendre au sens général et correspond à :

- des motifs parfaitement connus
- des formes géométriques
- des lettres, des chiffres
- des visages ...

- Reconnaissance d'objets d'intérêt

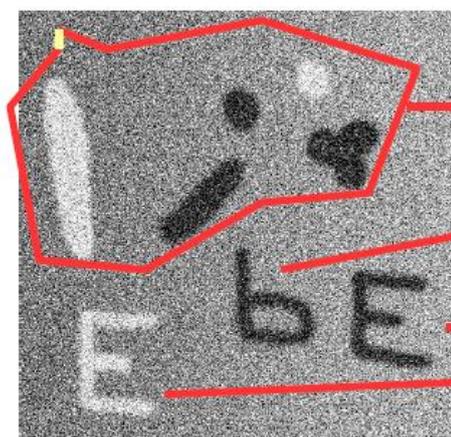
Exemple : reconnaissance de caractères

critère = forme (morphologie de l'objet)



Entrée : Image à analyser

Sortie : Résultat de l'analyse



Automatiquement associé à la classe « objets différents de lettres »

Automatiquement associé à la classe « lettre b »

Automatiquement associé à la classe « lettre e »

• Méthodes de reconnaissance

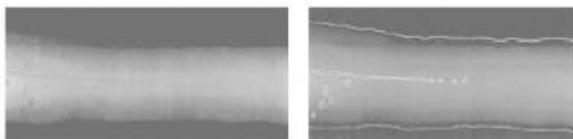
- On distingue principalement deux approches de la reconnaissance des formes :
 - les méthodes **statistiques** : l'extraction des caractéristiques produit des valeurs numériques qui sont confrontées aux modèles statistiques caractérisant chaque classe
 - les méthodes **structurelles** (ou **syntaxiques**) : l'extraction des primitives produit des valeurs symboliques et des relations qui font l'objet d'une analyse structurelle ou syntaxique

+ Applications

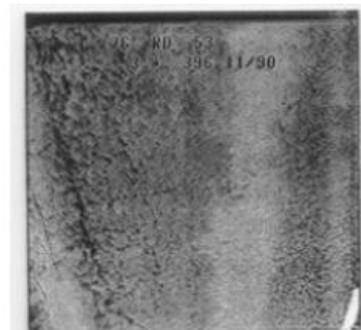
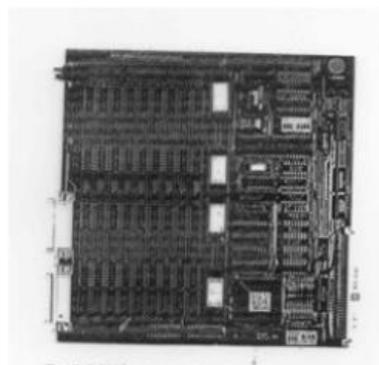
- Lecture de plaque d'immatriculation et plaques de signalisation
- Reconnaissance d'écriture manuscrite et signature
- Indexation d'images
- Détection de visage, d'objets, de panneaux...
- Détection de pièces défectueuses, contrôle de qualité industrielle
- Tri d'objets industriels, guidage de robots industriels

➤ Exemple d'applications: Détection de défauts

Image Rx d'une soudure: accepter ou rejeter?



Circuits en production: accepter ou rejeter?



• Processus d'analyse pour une RDF

1- Prétraitements

- ❖ binarisation, localisation, segmentation, élimination du bruit, normalisation, ...

2- Extraction des informations pertinentes

- **signal**: fréquence fondamentale, harmoniques, énergie, ...
- **image**: occlusions, concavités, contours, fins de trait, ... d'info.

3- Représentation de ces informations en vue de leur classification

- vecteur, graphe, chaîne

4- Classification de la forme

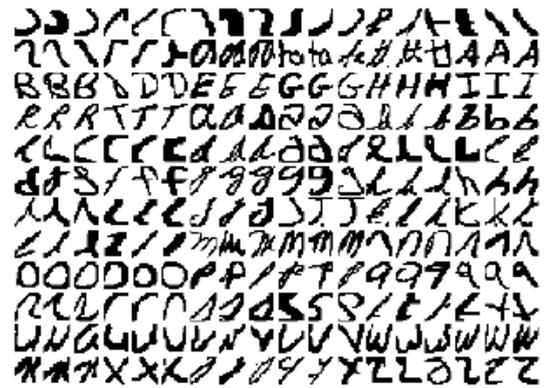
- apprentissage/décision, combinaison/fusion.

• Extraction de Caractéristiques

But: décrire la forme bidimensionnelle au moyen d'un ensemble de paramètres adéquats (attributs)

- **Large variété de représentations suivant les caractéristiques utilisées:**
 - Caractéristiques globales:
 - Surface, périmètre, largeur, hauteur
 - Elongation, circularité
 - Moments statistiques
 - Caractéristiques locales:
 - Coins ou sommets (nombre, positions relatives ou absolues, angles, ...)
 - Segments (nombre, positions relatives ou absolues, longueur, ...)
- **Les caractéristiques peuvent être extraites sur:**
 - la forme elle-même
 - son squelette
 - ses contours
- **Contraintes:**
 - faible variance intra-classe
 - grande variance inter-classe
 - indépendance en translation, rotation et facteur d'échelle
 - faible nombre d'attributs
- **Variabilité inter-classe vs variabilité intra-classe?**

0	0	0	0	0	C	σ	α	β	ο
1	1	1	1	1	2				
2	2	2	2	2					
3	3	3	3	3					
4	4	4	4	4					
5	5	5	5						
6	6	6							
7	7	7	7	7					
8	8	8	8	8	8	8			
9	9	9	9	9					
F	F	F	F						



• **Reconnaissance de formes par classification de caractéristiques**

- Reconnaître une forme, c'est prendre une décision sur la base de caractéristiques (ou descripteurs) identifiés.

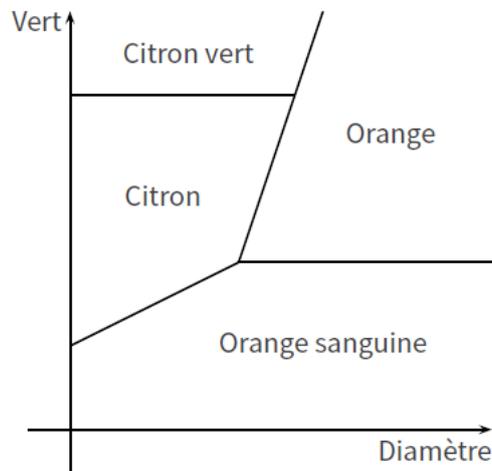
Exemples de caractéristiques : luminosité, couleur, taille, forme, orientation du gradient, ...

- Les caractéristiques sont des nombres, que l'on peut regrouper dans un vecteur de taille N et représenter dans un espace de dimension N .
- La décision est effectuée en effectuant une classification des caractéristiques dans cet espace.



- Classification de caractéristiques

Deux caractéristiques : diamètre et vert ($N=2$)



❖ Mise en place du système de reconnaissance

- Prétraitements éventuels (segmentation, filtrage...)

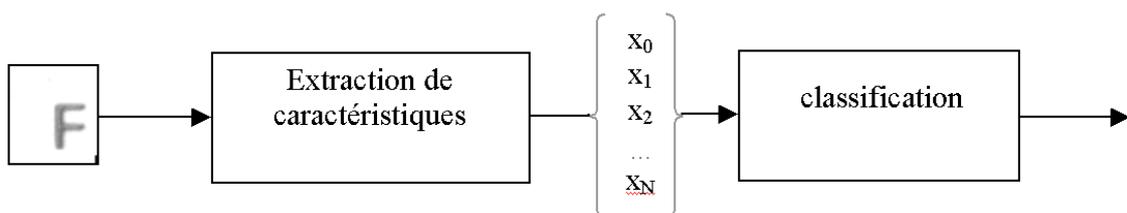
Définition de zones dans l'image (découpage en imagerie, fenêtre glissante, détection...)

- Détermination des caractéristiques (*features*) de ces zones
- Classification de ces caractéristiques à l'aide d'un classifieur (*classifier*) avec apprentissage (*learning*) préalable le cas échéant
- Évaluation des performances avec une base de test

• Classification de données pour la reconnaissance de formes

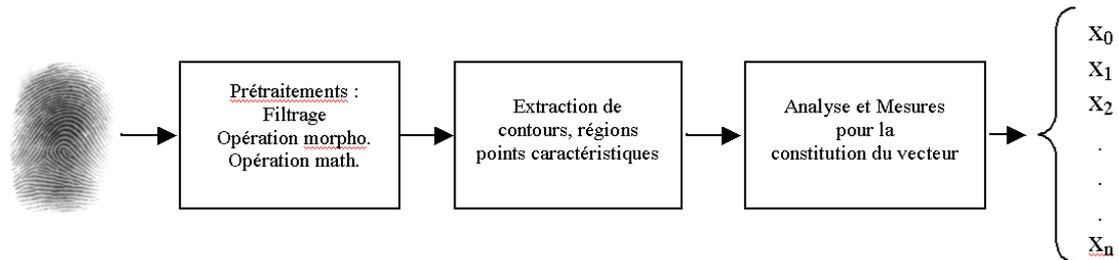
- Extraction de caractéristiques
- Prise de mesures sur l'objet
- Outils de classification
- Définition de critères sur les mesures

➤ Schéma générale



- **Extraction d'attributs (ou primitives)**
- **Mesures métriques, topologiques, statistiques**
- **Méthodes de classification**
 - Approche structurelle, syntaxique, statistiques

- **Extraction des attributs**

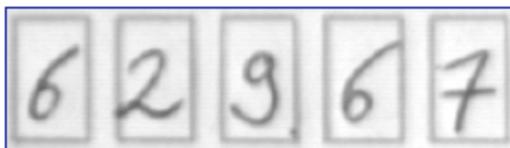


- **Mesures métriques**

- Mesure de distance
- Mesure de périmètre
- Calcul de surface
- Moments invariants
- Moments d'ordre 2
 - Largeur et hauteur de la forme
- Mesures invariantes
 - Compacité
 - Elongation

Exemple

Lecture d'un code postal



Code postal à lire



Binarisation



Traitement morphologique



Analyse en composante connexes

- **Rôle du classifieur** : le système doit être capable de déterminer si le vecteur :

- appartient à une classe
- n'est pas parfaitement conforme
- est à la frontière d'une classe
- n'appartient pas à des classes connues
- appartient à plusieurs classes
- ❖ impossible à traiter

- Les différentes approches

Approche structurelle

- Arbre ou graphe de décision
- Critère de distance

Approche syntaxique

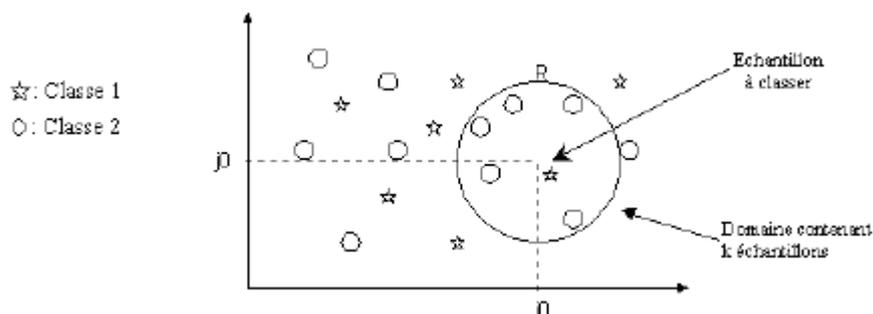
- Utilisation d'une « grammaire »

Approche statistique

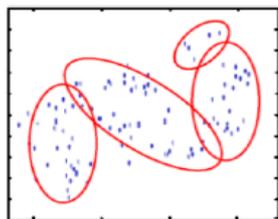
- Les plus utilisées de nos jours
- Basée sur la règle de Bayes

❖ Exemple

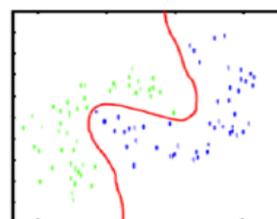
Règle du k-ppv : les plus proches voisins sont recherchés parmi k



❖ Apprentissage supervisé ou non?



Apprentissage non supervisé : faire des groupes à partir de données non étiquetées (clustering)

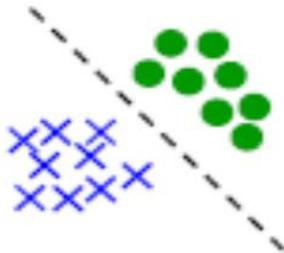


Apprentissage supervisé : trouver les fonctions séparant les classes (classification)

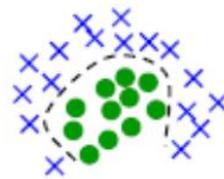
❖ Choix des caractéristiques

Un « bon » vecteur de caractéristiques est représentatif et discriminant :

- des échantillons issus de la même classe ont des caractéristiques similaires;
- des échantillons de classes différentes ont des caractéristiques différentes.



« bons » vecteurs



« mauvais » vecteur

Les performances des classifieurs se dégradent quand la dimension de l'espace des paramètres augmente : un grand nombre de caractéristiques ne permet donc pas forcément d'obtenir une bonne classification (*Curse of dimensionality*).

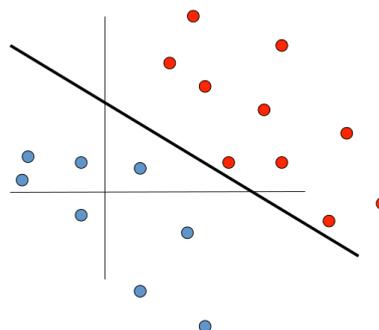
- utilisation d'un nombre restreint de caractéristiques pertinentes.

❖ Choix du classifieur

- Distance minimale
- k-moyennes (*k-means*)
- Machine à vecteurs de support (*support vector machine*)
- Forêt d'arbres décisionnels (*random forest*)
- Réseaux de neurones (*neural network*)

➤ Distance minimale

Les classes sont séparées par un hyperplan.



La frontière de décision entre deux classes i et j correspond à la perpendiculaire au segment reliant les moyennes de chaque classe, calculée lors de la phase d'apprentissage.

Les caractéristiques sont affectées à la classe dont la moyenne est la plus proche.

Avantage

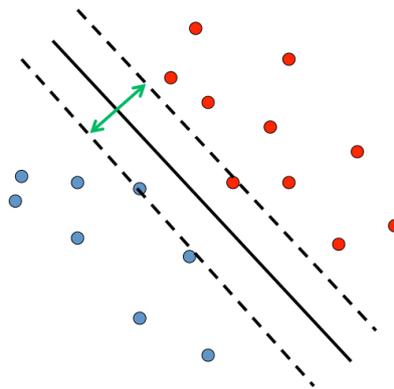
- Méthode extrêmement simple

Inconvénient

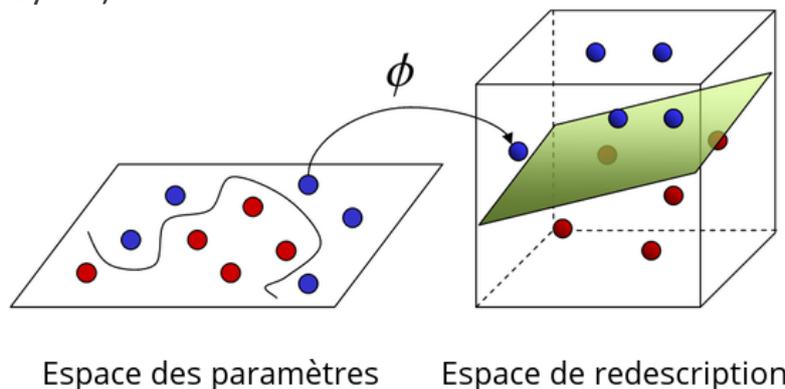
- Non applicable dans le cas où les données ne sont pas linéairement séparables

➤ Machine à vecteurs de support (SVM : support vector machine)

Les SVM sont un type de classifieur linéaire qui cherche l'hyperplan maximisant la marge entre les classes : optimisation.



Les éléments les plus proches de l'hyperplan sont appelés *vecteurs Supports*. Si les données ne sont pas linéairement séparables *on applique une transformation de l'espace des paramètres en un espace de plus grande dimension pour trouver une séparation linéaire dans ce nouvel espace (méthodes à noyaux).*



Avantages

- Nombreuses implantations disponibles
- Méthodes à noyaux très puissantes et flexibles
- Coût calculatoire réduit
- Peut être très performant, même sur un nombre restreint d'échantillons

Inconvénients

- Extension à plus de deux classes difficile (demande d'utiliser plusieurs SVM à deux classes)
- Difficulté de trouver un noyau optimal dans le cas non linéaire
- La phase d'apprentissage est coûteuse en temps de calcul

• Conclusion

Mise en place d'un système de reconnaissance de forme :

- prétraitement éventuel (segmentation, filtrage...)
- choix et calcul des caractéristiques pertinentes
- choix du classifieur
- entraînement & optimisation du classificateur (estimation de ses paramètres)
- évaluation des performances du classifieur