



Traitement d'images

Ch. 5 : Segmentation

s.idbraim@uiz.ac.ma

Master Spécialisé Offshoring des Technologies de l'Information
A.U. 2016 - 2017

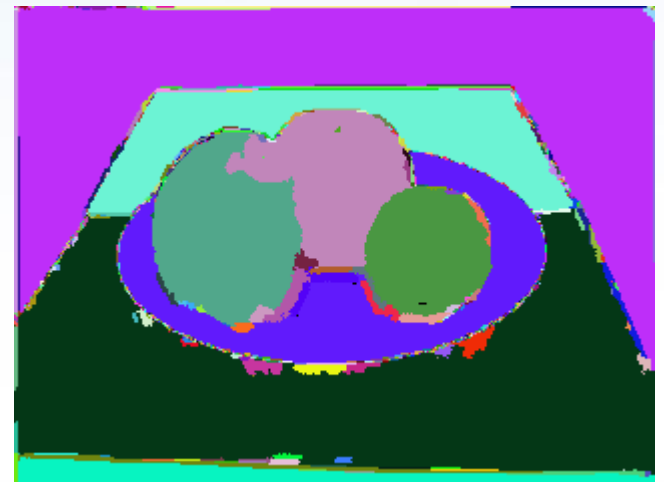
Qu'est-ce que la segmentation ?

- La segmentation vise à diviser l'image en morceaux

Ces morceaux correspondent aux objets dans l'image

- La segmentation est liée à la reconnaissance

Quels objets voit-on dans l'image ?



Qu'est-ce que la segmentation ?

- La segmentation est normalement basée sur:
 - *les discontinuités : contours*
 - les changements abruptes, frontières entre régions...
 - *les zones homogènes : régions*
 - Mêmes couleurs, textures, intensités, ...
- La segmentation est le découpage d'une image en différentes **régions** et/ou **contours**
- **Un contour fermé est équivalent à une région**

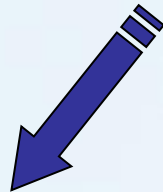
Segmentation régions/contours

Trois approches :

- **Approche « contours » :**
Les régions sont délimitées par les contours des objets qu'elles représentent (séparation)
- **Approche « régions » :**
Les régions sont déterminées en fonction de leurs propriétés intrinsèques (agrégation de pixels fonction d'un critère d'homogénéité)
- **Approches duales « régions et contours »**

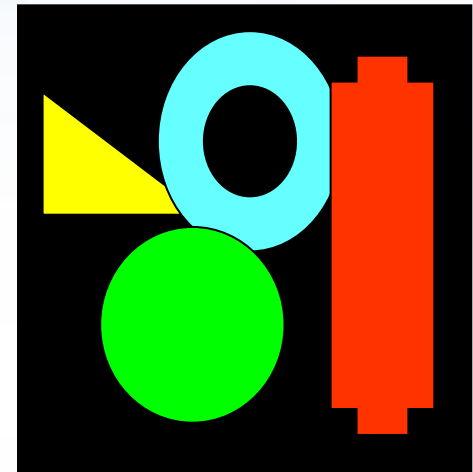
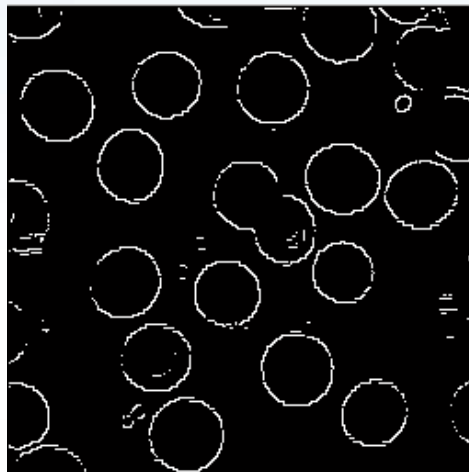
Segmentation régions/contours

Segmentation



Recherche de frontières
(approches « contours »)

Recherche de régions
(approches « régions »)



Segmentation par régions

- Méthodes statistiques:
 - ✓ Seuillage : de l'histogramme ...
- Méthodes géométriques
 - ✓ Split
 - ✓ Merge : Agrégation de pixels
 - ✓ Split and Merge
- Autres méthodes
 - ✓ Ligne de partage des eaux
 - ✓ Techniques d'optimisation markoviennes...

Segmentation par seuillage

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

avec

$$T = T[(x, y), p(x, y), f]$$

Coordonnées du pixel

Propriété locale au pixel

Image

Segmentation par seuillage

$$T = T[f]$$

Seuillage global

$$T = T[(x, y), p(x, y), f] \longrightarrow T = T[p(x, y), f]$$

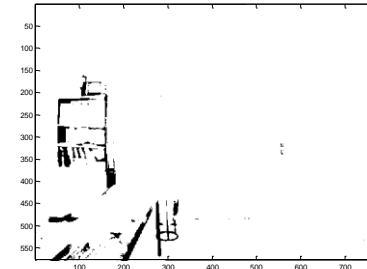
Seuillage local

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

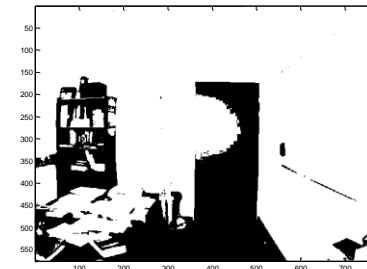
Seuillage dynamique

Segmentation par seuillage

Seuillage global



$$T = 64$$



$$T = 127$$



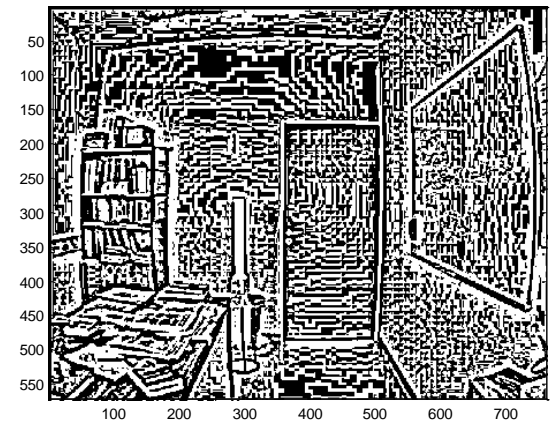
$$T = 180$$

Segmentation par seuillage

Seuillage local

Le seuil est la valeur moyenne d'une fenêtre 5x5 centrée en (x,y)

$$T = T[p(x, y), f(x, y)]$$



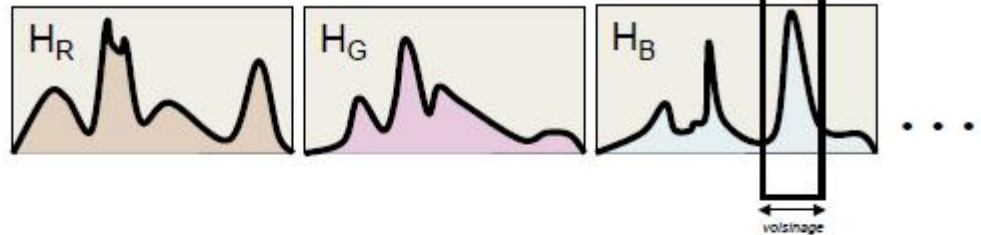
Segmentation par seuillage

Image couleur

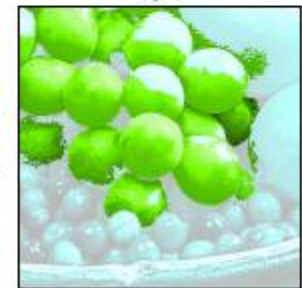
Chaque pixel est décrit selon certaines bandes : R,G,B,H,S,V,...
=> L'algorithme travaille sur plusieurs histogrammes, un par bande



Image initiale



Retroprojection
de la fenêtre de
l'histogramme



Suppression
de la région
extraite



Segmentation par seuillage

- Seuillage par minimisation de variance
- Seuillage entropique
- Seuillage par méthodes locales adaptatives
- ...

Segmentation par régions

Formulation

Soient une Image = Région R

Et P prédicat, critère, d'homogénéité

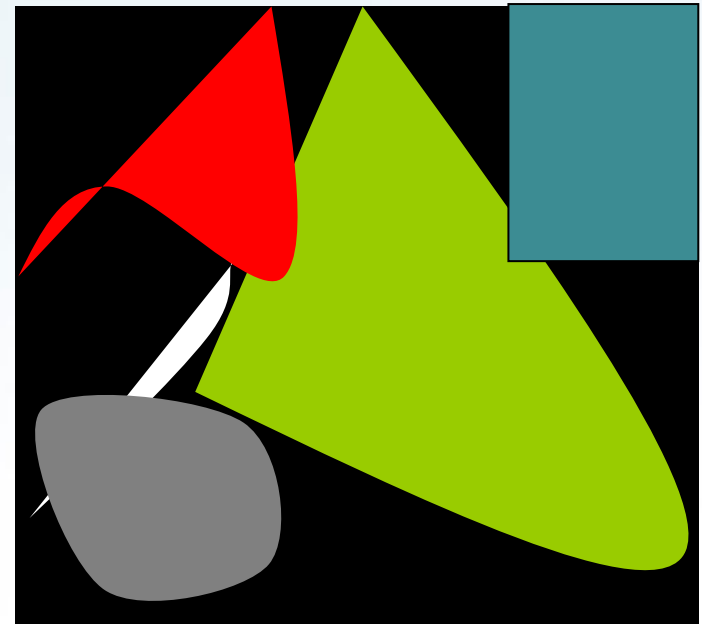
$$R = \bigcup_{i=1}^n R_i$$

R_i est connexe

$R_i \cap R_j = \emptyset$ pour $i \neq j$

$P(R_i) = \text{Vrai}$

$P(R_i \cap R_j) = \text{Faux}$ pour $i \neq j$



Segmentation par division (split)

Principe

- Définition d'un critère d'homogénéité
- Test de la validité du critère sur l'image
 - Si le critère est valide, l'image est segmentée [arrêt de la méthode]
 - Sinon, l'image est découpée en zones plus petites et la méthode est réappliquée sur chacune des zones

Segmentation par division (split)

Paramètres

- **Critère d'homogénéité** : valeurs identiques, variance limitée pour des paramètres de différentes nature
 - définition d'erreurs, de seuils ou d'intervalles de validité
- **Matière** : niveaux de gris, couleurs, textures, etc.
- **Décomposition de l'image** : division en 4, en 6, en polygones, etc.

Segmentation par division (split)

Exemple

0	1	0	0	7	7	7	7
1	0	2	2	7	7	7	7
0	2	2	2	7	7	7	7
4	4	2	2	7	7	7	7
0	0	1	1	3	3	7	7
1	1	2	2	3	7	7	7
2	4	3	0	5	7	7	7
2	3	3	5	5	0	7	7

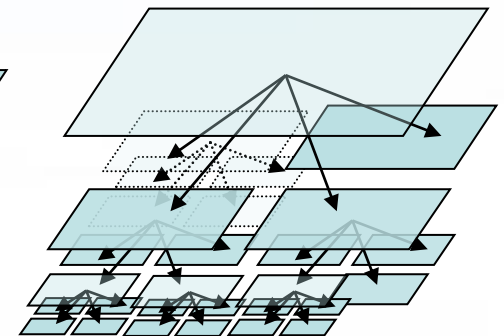
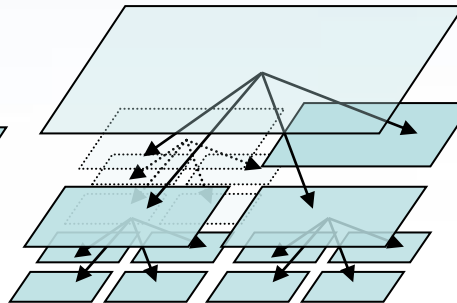
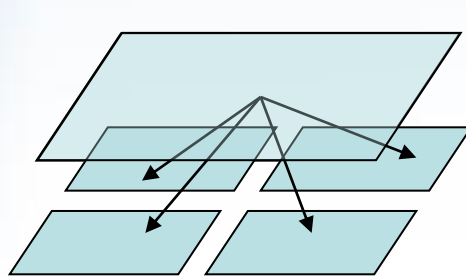
Split 1

0	1	0	0	7	7	7	7
1	0	2	2	7	7	7	7
0	2	2	2	7	7	7	7
4	4	2	2	7	7	7	7
0	0	1	1	3	3	7	7
1	1	2	2	3	7	7	7
2	4	3	0	5	7	7	7
2	3	3	5	5	0	7	7

Split 2

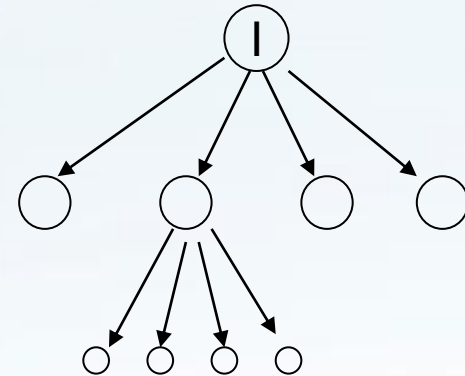
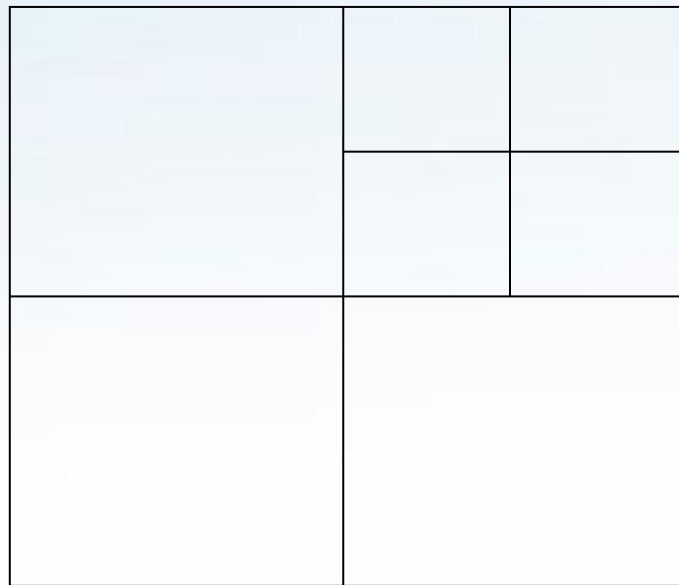
0	1	0	0	7	7	7	7
1	0	2	2	7	7	7	7
0	2	2	2	7	7	7	7
4	4	2	2	7	7	7	7
0	0	1	1	3	3	7	7
1	1	2	2	3	7	7	7
2	4	3	0	5	7	7	7
2	3	3	5	5	0	7	7

Split 3



Segmentation par division (split)

Quad tree



Segmentation par fusion (Merge)

Principe

- On explore l'image à partir de petites régions
- On fait croître celles-ci si elles satisfont à un critère d'homogénéité ou de regroupement

Paramètres

- Choix du critère d'homogénéité : différence de niveau de gris moyen, valeurs similaires, etc.
- Critère d'arrêt

Segmentation par fusion (Merge)

Méthode par agrégation (Croissance de régions)

Principe

on fait croître une région avant de passer à la suivante, sans parcours particulier déterminé *a priori* (méthode par agrégation libre de pixels)

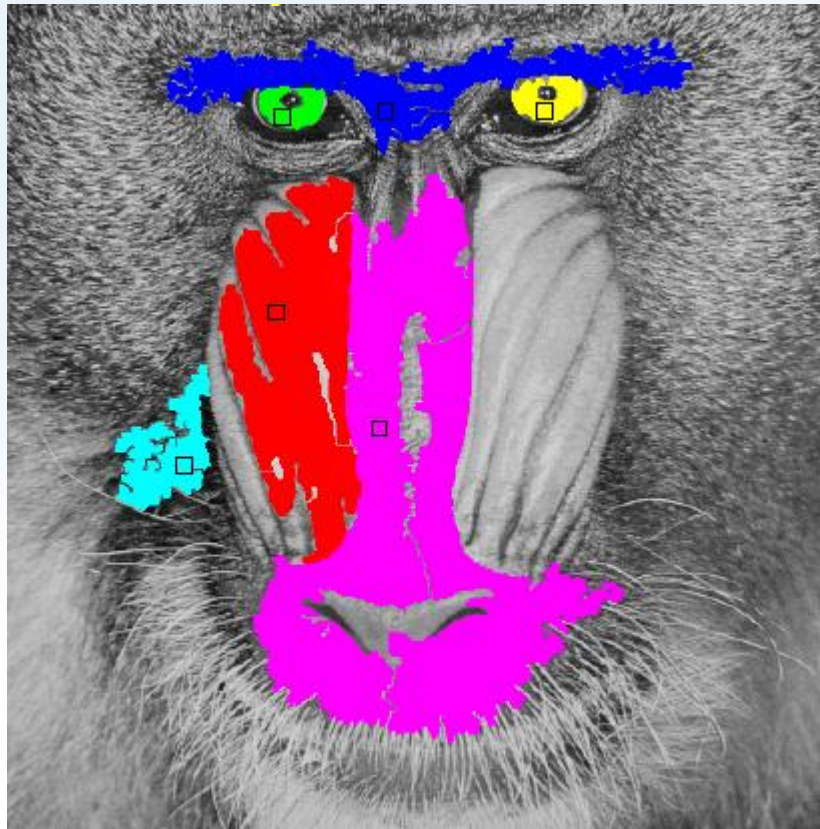
- Germe [*seed*] :
- Croissance suivant un critère de similarité
- Critère d'arrêt : convexité maximum, etc.

Inconvénients

- Méthode récursive, risques de débordements
- Influence de la position initiale du germe

Segmentation par fusion (Merge)

Méthode par agrégation (Croissance de régions)



Segmentation par Split and Merge

Proposition

Rassembler, à partir de la division grossière obtenue par **division**, les différents blocs adjacents de l'image

→ Algorithme de division et rassemblement, aussi appelé algorithme *Split and Merge*

Segmentation par Split and Merge

Mesures d'inhomogénéité

- Inhomogénéité régionale:

$$E(R_l) = \frac{1}{N_l} \sum_{n=0}^{N_l-1} [I(n) - \mu(R_l)]^2$$

- Prédicats d'uniformité:

$$P(A, R_l) = \begin{cases} \text{Vrai} & \text{si } E(R_l) < T_1 \quad (\text{tolérance}) \\ \text{Faux} & \text{sinon} \end{cases}$$

- Inhomogénéité interrégionale:

$$E(R_l, R_m) = \frac{1}{N_l + N_m} \sum_{n=0}^{N_l-1+N_m-1} [I(n) - \mu(R_l, R_m)]^2$$

Segmentation par Split and Merge

Critère de fin de rassemblement des régions

SQE(.) : somme des erreurs quadratiques

$$SEQ(L) = \sum_{i=0}^{L-1} \sum_{\substack{n=0 \\ n \in R_i}}^{N_i-1} [I(n) - \mu(R_i)]^2$$

Nombre de
Régions

Valeur du pixel
au point n

Moyenne des pixels
dans la région R_i

Critère d'arrêt : $SEQ(L) > T_2$ (valeur fixée *a priori*)

Segmentation par Split and Merge

Exemple

- L'image est tout d'abord divisée en choisissant $T_1 = 0,2$ comme valeur de seuil de tolérance de l'inhomogénéité régionale

$$E(R_l) = \frac{1}{N_l} \sum_{n=1}^{N_l-1} [I(n) - \mu(R_l)]^2 =$$

- Les différentes régions sont ensuite rassemblées en choisissant $T_2 = 5$ comme valeur de seuil appliqué au critère de fin de fusion (rassemblement)

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

Image originale

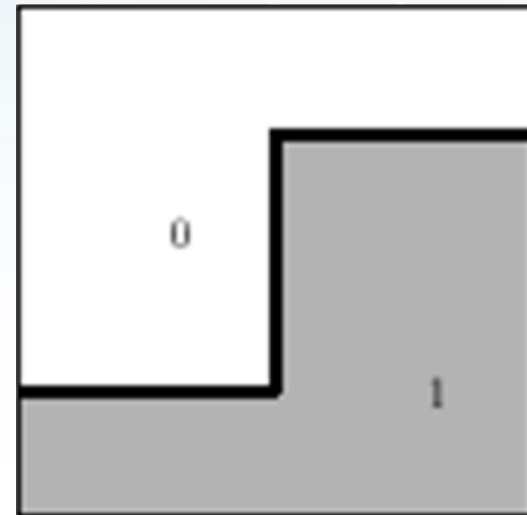
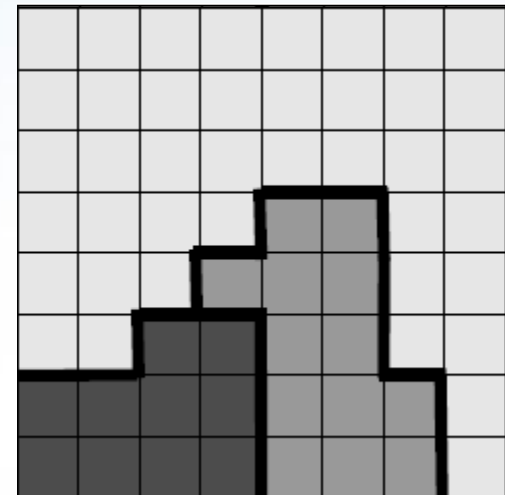
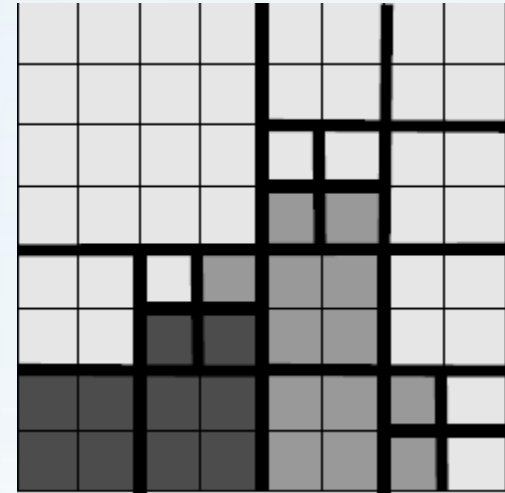
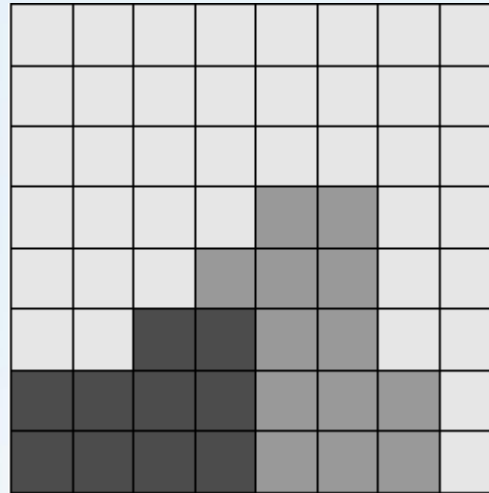


Image divisée $T_1 = 0,25$

Segmentation par Split and Merge

Exemple



Segmentation par Split and Merge

Exemple



Segmentation avec différents seuils

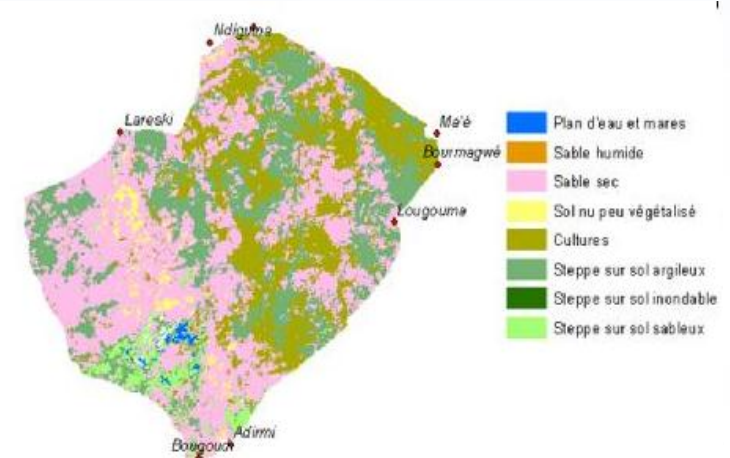
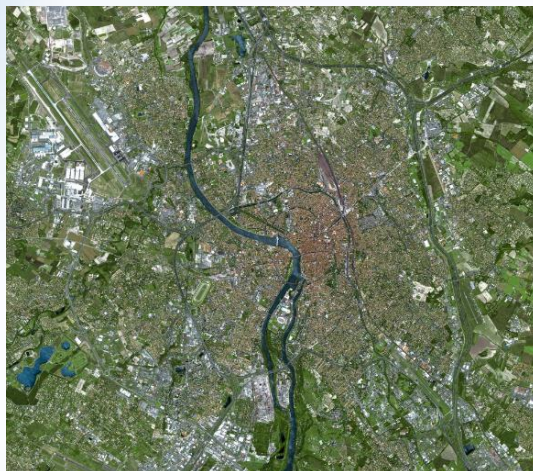
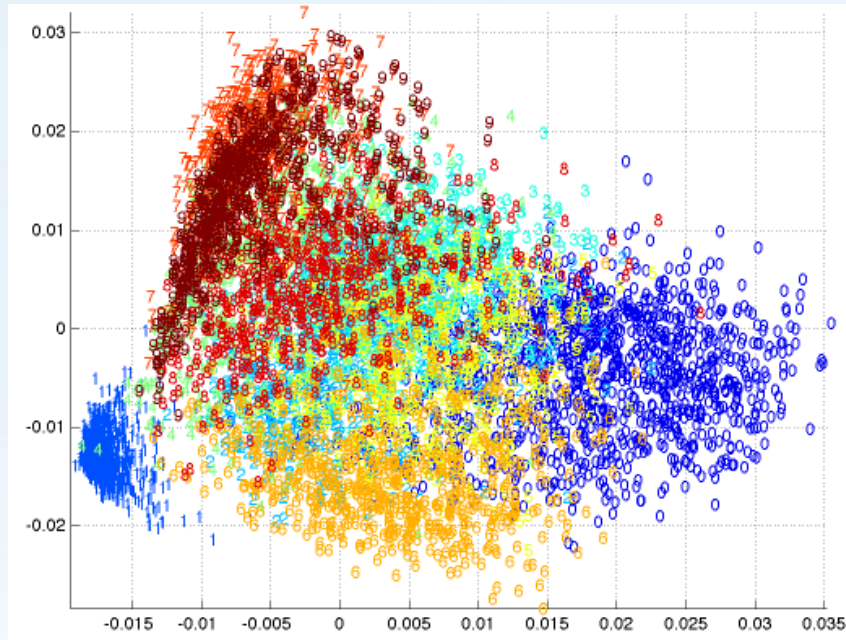
Segmentation par classification



Multitude de méthodes, dont :

- techniques de nuées dynamiques (K-means, ISODATA...)
- réseaux de neurones
- hyperplans/hypersurfaces séparateurs (SVM)
- champs de Markov
- Lignes de partage des eaux (LPE)
- ...

Segmentation par classification



Segmentation par contours

Du gradient au contour

- Extraction des points de contours
 - La phase d'extraction se termine par un seuillage
 - ☞ introduit des manques, dus à la présence de bruit ou à des occultations
- Suivi de contours
 - Fermeture des contours très importante en segmentation :
 - ☞ Permet d'obtenir des régions fermées interprétables comme projections des objets de la scène

Segmentation par contours

Les approches de fermeture les plus courantes :

- Exploitation des techniques d'exploration des graphes (IA, prog. dynamique)
- Méthodes neurométriques (réseaux de neurones)

Principe des méthodes basées sur les techniques d'exploration des graphes

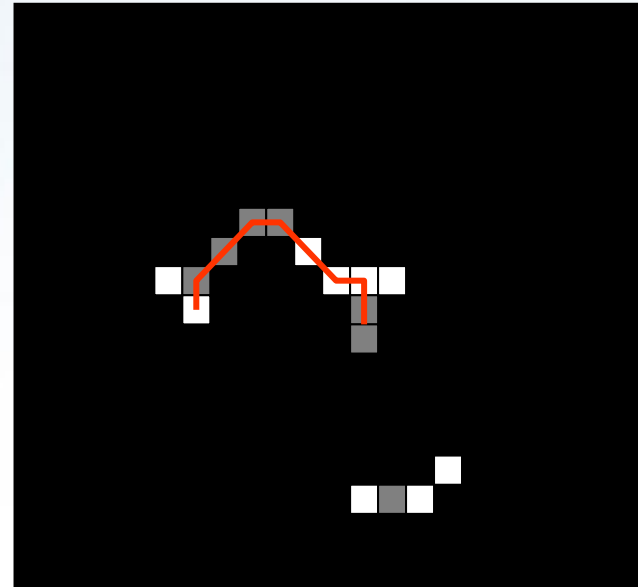
- On part des point possibles des contours ($\rho(m,n) > s_h$)
- On identifie les extrémités des chaînes des points de contour
- Analyse de la configuration de voisinage
- On cherche à propager ces extrémités :

Segmentation par contours

Pour chaque pixel (x,y) d'un contour (fort gradient), on cherche les pixels adjacents (x',y') dont le gradient est peu différent de celui de (x,y) .

$$|\nabla f(x, y) - \nabla f(x', y')| < S_{Module}$$

$$|a(x, y) - a(x', y')| < S_{Angle}$$



Conclusion

La segmentation d'une image cause encore aujourd'hui beaucoup de problèmes

- *Aucune méthode ne fonctionne pour toutes les images*
- *Pas de garantie, pas de recette miracle !*

Le **pré-traitement** des images, la **sélection de capteurs** et **sources d'énergie** appropriées, et la **prise contrôlée des images** rendent cette étape plus facile et plus efficace

La segmentation aide beaucoup pour la reconnaissance, mais elle n'est pas obligatoire dans tous les cas

- *Voir cours de Vision par Ordinateur*

Conclusion

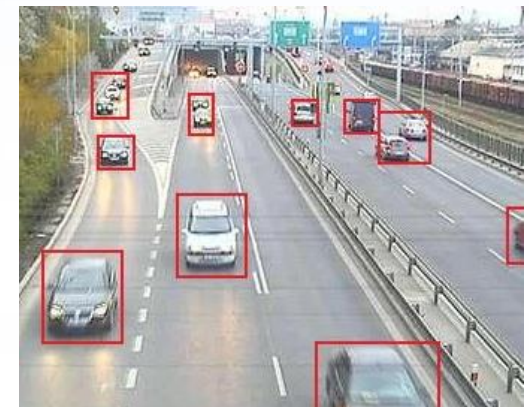
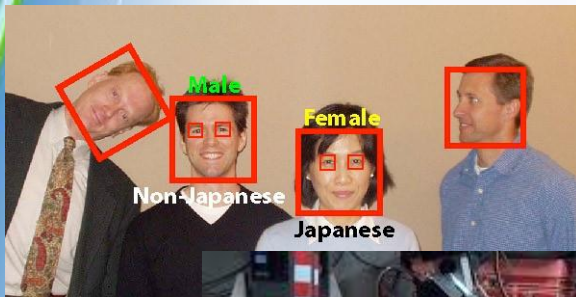
Evaluer le résultat d'une segmentation n'est pas facile

- *Il dépend de l'application*
- *Il dépend de ce qu'on veut*
- *Il est subjectif et varie d'une personne à l'autre*

Un des principaux problème est de définir le but de la segmentation

Qu'est-ce qu'on recherche exactement dans l'image ?

Segmentation de visages, segmentation sur des patterns de mouvement ...





Fin