



Image, Modélisation et Rendu

Parcours Multimédia, Département SN

Parcours IATI, Département 3EA

Des transformations à l'analyse d'images

Partie II

Sylvie CHAMBON
schambon@enseeiht.fr

16 février 2024



Segmentation

Introduction

Approches par contours

Approches par régions

Approches par classification

Conclusion

Superpixels

Introduction : origine

Définition et propriétés

Construction



Plan de la présentation

Segmentation

Superpixels



Introduction : Première définition

Partition de l'image sachant que chaque partition représente une information homogène et cohérente de la scène qui a du sens

Information cohérente = ressemblance/similarité/faible différence sur un critère donné

- Autant de définitions que de manière de définir qu'une information est homogène et cohérente
- Problème de segmentation mal posé
- Choix d'une méthode lié à
 - Nature des images (éclairage, texture, ...)
 - Forme des primitives à extraire :
couleur, niveau de gris ou plus généralement, la photométrie, mais aussi la texture, les contours ...
 - Contraintes de temps



Introduction : Illustration



Image originale



Image segmentée

Distinguer des objets



Image originale

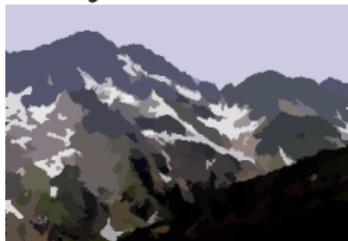


Image segmentée

Distinguer des éléments naturels



Introduction : Domaines

Dans la littérature, ces deux catégories d'approches sont souvent distinguées :

- Segmentations fond/forme (2 classes seulement)
- Segmentations à plusieurs objets

Nous distinguons également

- **Segmentation supervisée** : base d'apprentissage pour avoir des connaissances *a priori* sur les régions recherchées
- **Segmentation semi-supervisée** : seule connaissance *a priori* = le nombre de régions recherchées
- **Segmentation non supervisée** : aucune connaissance *a priori*



Introduction : Définition formelle

- Créer une partition de l'image I en sous-ensembles R_i appelés régions tels que :

$$\forall i R_i \neq \emptyset$$

$$\forall i, j \text{ avec } i \neq j R_i \cap R_j = \emptyset$$

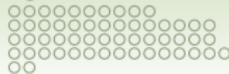
$$I = \cup_i R_i$$

- **Une région** = ensemble de pixels **connexes ou non** ayant des propriétés communes qui les différencient des régions voisines : niveaux de gris/couleurs, textures, formes



Introduction : Classement des méthodes

- Classiquement
 - **Approches contours** contours entre objets peu semblables ou entre zones hétérogènes
 - **Approches régions** pixels ressemblants pour former des régions homogènes
 - **Approches par classification** approches bayésiennes, approches par Espérance-Maximisation (EM), *k*-moyennes (*k-means*), *mean-shift*
 - **Approches hybrides ou duales** mélangeant les deux aspects
- Plus récemment
 - **Approches par segmentations multiples**
 - **Approches par sur-segmentation par superpixels** première estimation fiable de la segmentation finale
 - **Approches par combinaison des informations de nature différente** informations 3D ou informations contextuelles



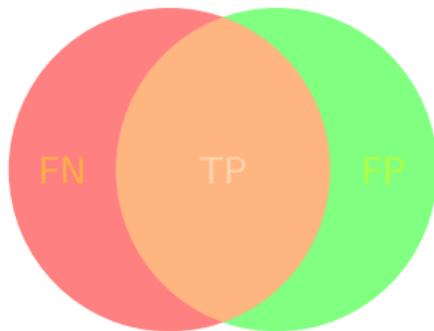
Introduction : Évaluation de la qualité du résultat

Ce que l'on a :

- La classe segmentée (à évaluer)
- La classe à segmenter (référence)

Ce que l'on évalue :

- Vrais positifs, *True Positives* (TP) (en blanc)
- Faux positifs, *False Positives* (FP)
- Faux négatifs, *False negatives* (FN)
- Vrais négatifs, *True Negatives* (TN)





Introduction : Évaluation de la qualité du résultat

- Précision P
- Proportion de vrais positifs parmi les positifs (la segmentation estimée) :

$$P = \frac{TP}{TP + FP} \quad (1)$$



Introduction : Évaluation de la qualité du résultat

- Sensibilité S
- = Proportion de bonnes détections par rapport à la forme à segmenter
- = **Proportion de manques**

$$S = \frac{TP}{TP + FN} \quad (2)$$



Introduction : Évaluation de la qualité du résultat

- Coefficient de similarité
- ou *DICE Similarity Coefficient*
- ou Mesure/Score f_1

DSC

= Moyenne harmonique entre la précision et la sensibilité :

$$DSC = \frac{2TP}{2TP + FP + FN} \quad (3)$$



Plan de la présentation

Segmentation

Superpixels



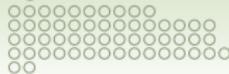
Des approches classiques aux contours actifs

- Approches utilisées anciennement : basée détection de contours, fermeture



Des approches classiques aux contours actifs

- Approches utilisées anciennement : basée détection de contours, fermeture
- Approches utilisées plus récemment : contours actifs



Des approches classiques aux contours actifs



Nous souhaitons segmenter les grains de riz et les poivrons

- Colonne 2 : les contours avec fermeture
- Colonne 3 : ajout d'un label en niveau de gris pour chaque région dans l'ordre des classes détectées et des couleurs possibles : effet de dégradé
- **Détection de contours versus segmentation basée contours**



Des approches classiques aux contours actifs



Nous souhaitons segmenter les grains de riz et les poivrons

- Colonne 2 : les contours avec fermeture
- Colonne 3 : ajout d'un label en niveau de gris pour chaque région dans l'ordre des classes détectées et des couleurs possibles : effet de dégradé
- **Détection de contours versus segmentation basée contours**



Principe des contours actifs

Définition subjective et intuitive de l'énergie :

- Application à un contour : soit l'ensemble des pixels qui le compose
- Mesure d'adéquation entre le contour choisi et les propriétés attendues pour ce contour
- **analogie avec l'énergie : mesure la quantité d'énergie qu'il va falloir dépenser pour répondre aux propriétés attendues**



Principe des contours actifs

- **Hypothèse** : Un contour idéal pour segmenter la forme
- **Objectif** : Estimer ce contour idéal
- **Caractéristique importante** : Algorithme itératif partant d'un contour initial et évolution en utilisant la notion d'énergie
- **Principe** : Déterminer l'ensemble des pixels qui forment le contour, tout en minimisant un critère attendu.

critère = énergie

- **Rôle de l'énergie** : Savoir lorsque le contour estimé est le plus proche du contour idéal
- **Lien contour/énergie**
 - Toute règle ou contrainte appliquée ou ajoutée à l'énergie
= directement ou indirectement une règle ou une contrainte sur le contour lui-même

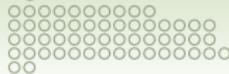


Principe des contours actifs

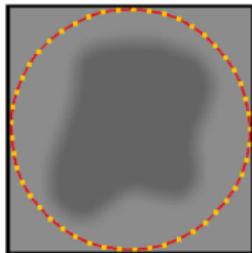
Explications des termes de l'énergie

Caractéristiques des contours actifs

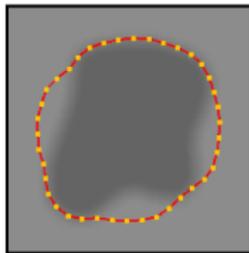
- Énergies pour représenter **propriétés physiques** et **adéquation à la forme**
- **Énergie élastique** pour préciser la configuration souhaitée (information *a priori* sur la forme)
- Attribution d'une inertie pour un **comportement dynamique** (connaissances *a priori* sur le mouvement)
dans la séquence de déformation



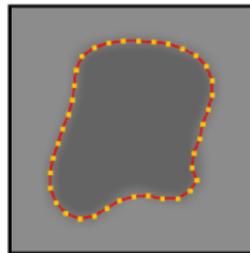
Étapes des approches par contours actifs



Initialisation



Évolution

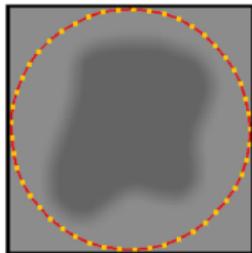


Convergence

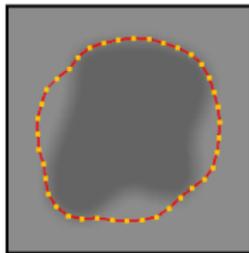
1. Initialisation du contour : **Forme simple, détection de contours, interaction humaine**



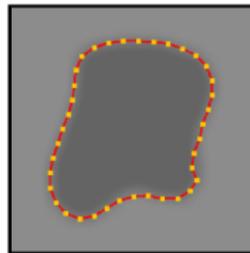
Étapes des approches par contours actifs



Initialisation



Évolution

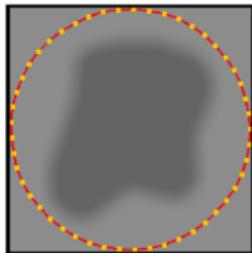


Convergence

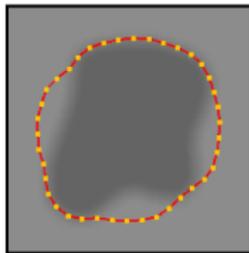
1. Initialisation du contour : **Forme simple, détection de contours, interaction humaine**
2. Déformation par itérations successives jusqu'à convergence : introduction énergie E et son évolution



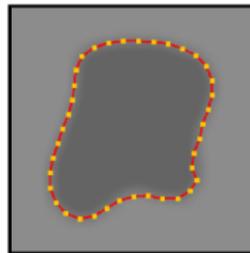
Étapes des approches par contours actifs



Initialisation



Évolution



Convergence

1. Initialisation du contour : **Forme simple, détection de contours, interaction humaine**
2. Déformation par itérations successives jusqu'à convergence : introduction énergie E et son évolution

Commentaires

- Première étape influence la qualité du résultat
- Contour éloigné de celui recherché = algorithme coûteux



Énergie à minimiser (1)

- Contour représenté dans le plan par une forme paramétrique fermée ou non fermée, notée :

$$\mathcal{C}(s) = (x(s), y(s)), s \in [0, 1]$$



Énergie à minimiser (1)

- Contour représenté dans le plan par une forme paramétrique fermée ou non fermée, notée :

$$\mathcal{C}(s) = (x(s), y(s)), s \in [0, 1]$$

- Forme générale de l'énergie E à minimiser :

forces internes (géométrie du contours)/forces externes (hors géométrie du contour)

$$E(\mathcal{C}) = \int_0^L \left(\underbrace{w_1 \|\mathcal{C}'(v)\|^2 + w_2 \|\mathcal{C}''(v)\|^2}_{\text{Forces internes}} + \underbrace{\mathcal{P}(\mathcal{C}(v))}_{\text{Force externe ou image}} \right) dv.$$

$[0, L]$, domaine de définition de \mathcal{C}

L , longueur de la courbe



Énergie à minimiser (2)

$$E(\mathcal{C}) = \int_0^L \left(\underbrace{w_1 \|\mathcal{C}'(v)\|^2 + w_2 \|\mathcal{C}''(v)\|^2}_{\text{Forces internes}} + \underbrace{\mathcal{P}(\mathcal{C}(v))}_{\text{Force externe ou image}} \right) dv.$$

- **Forces internes = aspect régulier de la courbe**

avec w_1, w_2 les poids de chaque élément

- w_1 = coefficient d'**élasticité**, résistance à l'allongement de la courbe
Pénalisation de la longueur de la courbe : **dérivée première**
- w_2 = coefficient de **flexibilité**, pénalité sur la courbure du contour :
dérivée seconde
- Choix délicat pour w_1 et w_2 dépendant de la forme recherchée : plus ou moins long, plus ou moins rigide
- **Force externe ou image**, *a priori* de forme = potentiel d'attraction ou énergie d'attache aux données



Choix possibles

- **Choix pour \mathcal{P} , la force externe**

1. *Basé niveaux de gris* : $\mathcal{P}(C(v)) = I(v)$
2. *Basé gradients*: $\mathcal{P}(C(v)) = \phi(\|\nabla I(v)\|)$ avec ϕ fonction décroissante

- **Type de contours**

- *Contours géométriques* : évolution planaire de la courbe
- *Contours géodésiques* : cas particulier des contours actifs avec

$$E(C) = \int_{\omega} \left(\|C'(v)\|^2 + \mathcal{P}(C(v)) \right) dv$$

contour géodésique = distance la plus courte entre deux points en prenant en compte, c'est-à-dire en suivant, le contour (en 3D, la forme de la surface) qui relie les deux points



Comment choisir ces différents paramètres

Propriétés attendues ?

- Pour la force interne, si nous imaginons que le contour initial à la bonne longueur, il faut imposer une longueur constante.
- Pour la force interne, si le contour doit être régulier, il faut imposer une régularité.
- Pour la force externe, si nous pensons que c'est la couleur qui apporte l'information la plus pertinente, il vaut mieux la favoriser.
- Pour la force externe, même chose pour les gradients ou la texture.
- Exemple : une fleur sur un fond végétal



Forme discrète

- Définition **en continu**
- Application en discret
- Définition d'une courbe avec des points de contrôle
- Choix d'un **pas d'échantillonnage**
- Choix de la forme : polygonale, courbe (B-spline, cf. cours interpolation)
- Choix de la méthode d'optimisation pour minimiser l'énergie : descente de gradient, recuit simulé



Algorithme basique de calcul de contours actifs

1. Calculer l'énergie pour chaque point de la courbe
2. Faire la liste des points par énergies croissantes
3. Tant que la condition d'arrêt n'est pas atteinte faire
 - 3.1 Faire évoluer le point d'énergie maximale :
déplacement dans son voisinage
 - 3.2 Attribuer la nouvelle énergie
 - 3.3 Si la distance est trop importante entre deux points,
ajouter un point



Conclusion sur les contours actifs

1. Contour trop éloigné : intervention humaine

2. Image bruitée ?

Lissage ok

Mais Risque de dégrader le calcul des dérivées et des gradients

3. Plusieurs objects ?

OK

Mais Plusieurs initialisations



Plan de la présentation

Segmentation

Superpixels



Approches par régions

- Approches par histogramme
- Division/Fusion
- Croissance de régions
- Ligne de partage des eaux



Introduction sur les approches par régions

- Les plus anciennes, traitements bas niveaux, de type seuillage
- Deux catégories distinctes
 1. **Approches par histogramme**

Défaut : pas toujours de cohérence spatiale
 2. **Croissance de régions et fusion division**

Défaut : Difficulté de définir un prédicat qui permet de bien séparer les classes



Approches par histogramme

- **Commentaires**

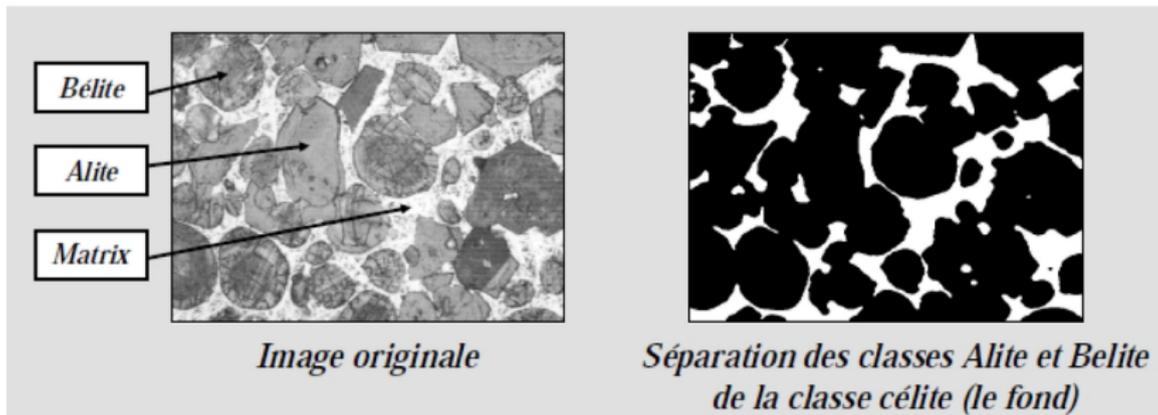
- les plus simples
- les plus adaptées dans le cas à plusieurs dimensions (hyperspectrales, couleurs)

MAIS

- sensibles au bruit
- **Deux types d'approches** : semi supervisées et non supervisées (ou automatique)
- **Seuillage (grâce à l'histogramme)** : isoler des pics de l'histogramme et effectuer un seuillage des zones correspondantes, cf. partie I



Seuillage de l'histogramme : Exemple d'application





Division/Fusion

- **Principe** : Diviser ou agréger les régions suivant un prédicat, de manière itérative jusqu'à une certaine convergence
- **Algorithme simple**
 - Soit R_i , issue d'une première segmentation par division
Cette segmentation initiale peut être une simple grille régulière de l'image
 - Soit $P(R_i)$ un prédicat logique donnant vrai si la région satisfait un certain critère (homogénéité)
 1. Diviser en régions chaque R_i telle que $P(R_i)$ est faux
 2. Fusionner toutes régions R_j et R_k telles que $P(R_j \cup R_k)$ est vrai
 3. Arrêter dès que la fusion/division est impossible
- **Intérêt** : Simple et peu coûteuse lorsque les régions sont bien séparables



Division/Fusion

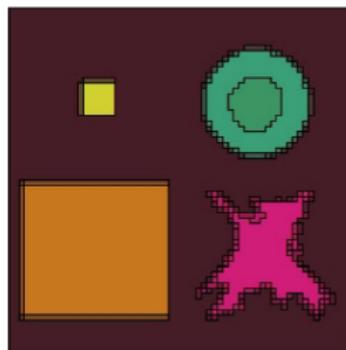
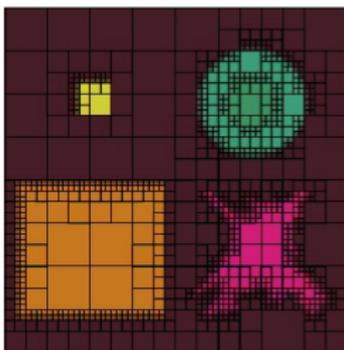
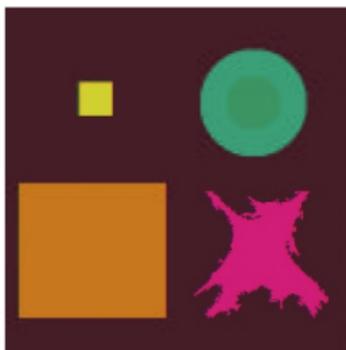
Détails à fixer

- Comment enchaîner les divisions/fusions (parfois toutes les divisions puis, toutes les fusions, ou fusions/divisions simultanées)
- Comment diviser les régions (souvent en 4)
- **Prédicat de fusion/division : On peut les combiner !**
 - Contraste : $P(R_i) = |\max(R_i) - \min(R_i)| < S$
 - Différences limitées : $\forall(x, y)$ pixels voisins $|I(x) - I(y)| < S$
 - Écart-type, moyenne, nombre de pixels, contour/périmètre ...
 - **Complexité de cet algorithme : $\log_2 N$ (N nombre de pixels)**



Division/Fusion

Initialisation, étape intermédiaire et résultat final



- Des divisions/fusions qui s'adaptent à la finesse des détails
- Chaque ordre de fusion/division va donner un résultat différents



Croissance de régions

- **Principe**

1. Sélection d'amorces : "cœur des régions"

Choix des amorces ? : zone homogène et/ou éloigné d'un contour

2. Pour chaque amorce **a** de coordonnées (x_A, y_A) :

Propager l'amorce en agglomérant tous les pixels **p** tels que :

$$|I(x_A, y_A) - I(x, y)| < S$$

- Propagation séquentielle ou parallèle (pour ne pas privilégier une région)
- Critère de propagation simple mais il peut être adapté : gradient, texture



Croissance de régions

- **Détails à fixer**
 - Choix des amorces (automatique ou non)
 - Choix du seuil
 - Choix de la propagation
 - **Séquentielle** = Chaque germe propagé entièrement l'un après l'autre
 - **Parallèle** = Tous les germes propagés en même temps
 - Choix de l'ordre dans lequel sont ajoutés les pixels dans une région
- **Commentaires** : implémentation relativement simple et temps d'exécution rapides
- Lorsque la propagation est en parallèle (dans le cas de multiples amorces), l'ordre dans lequel sont ajoutés les pixels dans une région a une influence sur le résultat. Pas d'intérêt, mais c'est un fait.



Ligne de partage des eaux

● Principes

- Description des images en termes géographiques
- Image = relief où le niveau de gris soit assimilé à une altitude/une profondeur
- **Ligne de partage des eaux = crête formant la limite entre deux bassins versants**
- Pour l'obtenir : immersion du relief dans l'eau (l'eau ne peut pénétrer dans les vallées que par ses minima)
- **Ligne de partage des eaux = points où deux lacs disjoints se rejoignent au cours de l'immersion**
- **Difficulté** : Mise en œuvre laisse beaucoup de liberté de formalisation, risque de sur-segmentation
- **Prise en compte de ces difficultés** : pré-filtrage, cf. partie I.
- **Intérêt** : À utiliser plutôt pour une seconde étape de segmentation



Ligne de partage des eaux

Une approche itérative

Mise en œuvre : détails à choisir

- Grille pour sélectionner des amorces ou Choix des **minima**/plateau

Exemples **Minima locaux en terme de niveau de gris ou gradient**

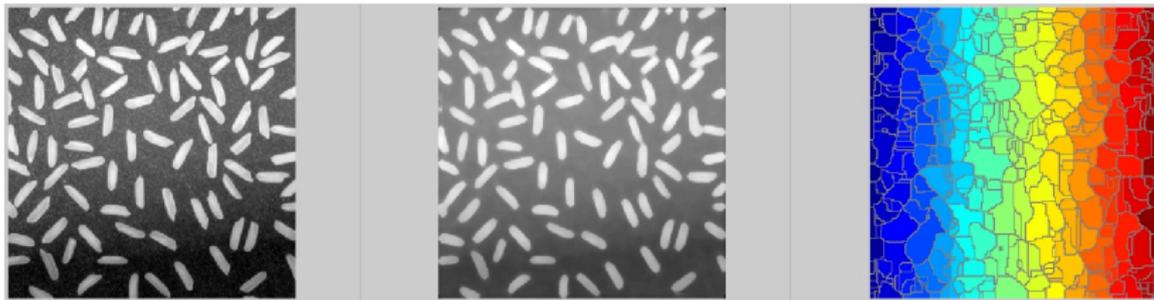
- Construction des bassins : par les chemins **minimaux** (difficiles à implémenter) ou par immersions

Exemples **Théorie des graphes ou croissance de régions**

- Approches itératives pour fournir une étiquette à tous les pixels



Ligne de partage des eaux : résultats



Quelles conclusions sur ces résultats ? Comment les utiliser ? Pourquoi cela ne fonctionne qu'avec des régions lisses ?



Ligne de partage des eaux : résultats



Quelles conclusions sur ces résultats ? Comment les utiliser ? Pourquoi cela ne fonctionne qu'avec des régions lisses ?



Ligne de partage des eaux : résultats



Quelles conclusions sur ces résultats ? Comment les utiliser ? Pourquoi cela ne fonctionne qu'avec des régions lisses ?



Ligne de partage des eaux : résultats combinés



(a)



(b)



(c)



(d)



Plan de la présentation

Segmentation

Superpixels



Approches par classification

- *k*-moyennes
- Mean-Shift



k -moyennes

- **Principe** : Connaissant le nombre de classes N , on cherche la meilleure répartition des pixels.
- **Algorithme itératif**
 1. Découper l'image en N régions suivant un critère c : utilisation d'une grille, à l'initialisation
 2. Estimer les centres C_k , $k \in [1, N]$, de ces régions
 - item Tant que les centres sont modifiés, faire
 - 2.1 Pour chaque pixel de l'image, trouver le centre C_{k^*} le plus proche, au sens du critère c , et donner au pixel considéré l'étiquette k^*
 - 2.2 Mettre à jour les centres C_k des N régions ainsi constituées
- Pas de preuve de convergence de cet algorithme vers l'optimum global
- Résultat fortement dépendant de l'initialisation



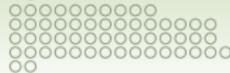
k -moyennes

Qu'est ce que le critère c ?

- Différence colorimétrique, aspect photométrique
- Distance euclidienne entre le centre et le point considéré, aspect spatial ou géométrique

Remarques

- Régions obtenues pas toujours connexes, cela dépend de ce qui est pris en compte dans le calcul des centres
- Variantes de cet algorithme : *mean-shift*



k-moyennes : illustration des itérations



image



1



2



3



4



5



6

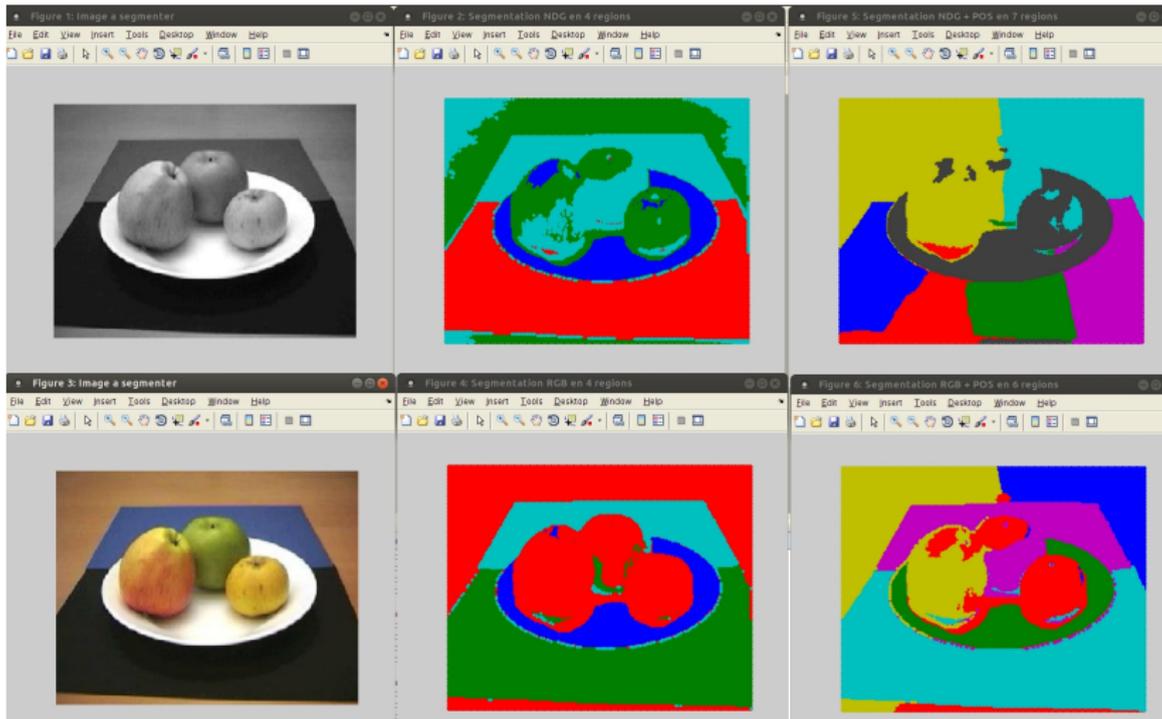


résultat



Approche classification

k-moyennes : illustration





Mean-Shift

- **Historique**

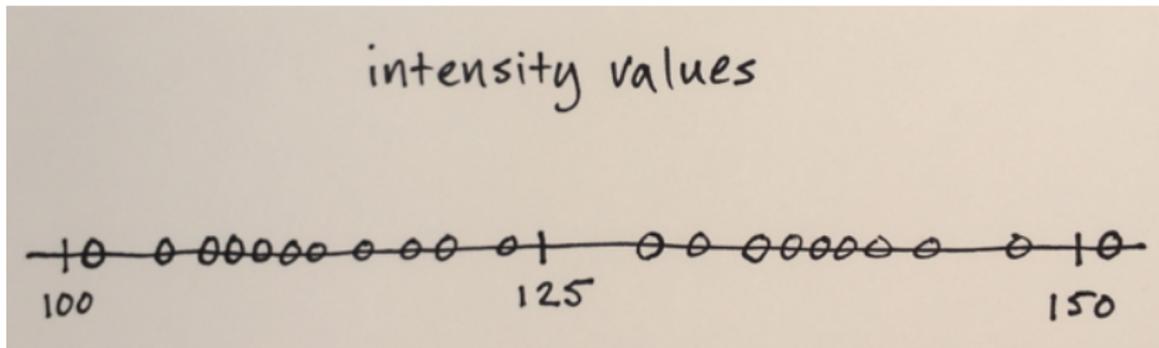
- Proposé en mathématiques pour **estimer les modes principaux d'une fonction**
- Repris par Comaniciu et Meer en segmentation d'images
- Avantage : non-supervisé

- **Principe**

- Estimer les modes d'un nuage de points de \mathbb{R}^d
- *mode* = **maximum local de sa fonction de densité**
- Outil utilisé : “**suivre**” la **direction du gradient** de la fonction de densité

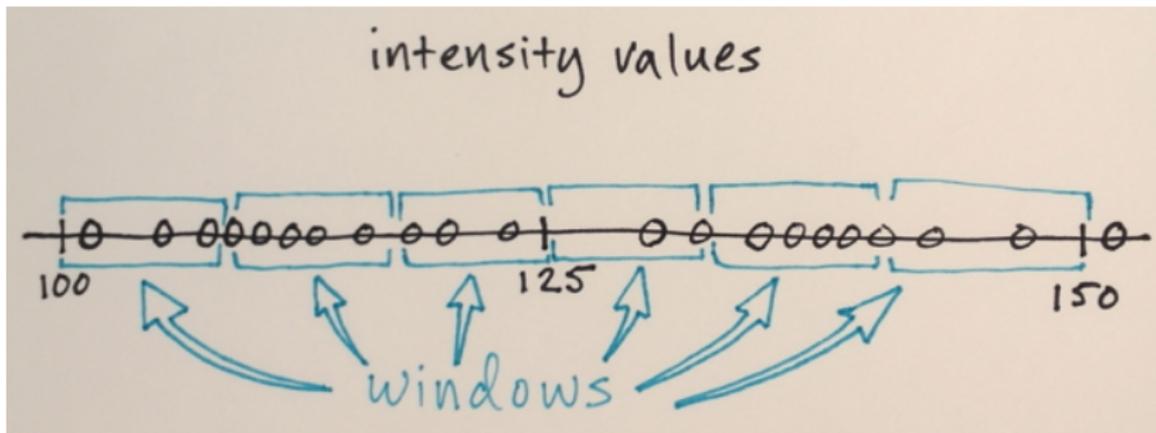


Mean-Shift : illustration du principe



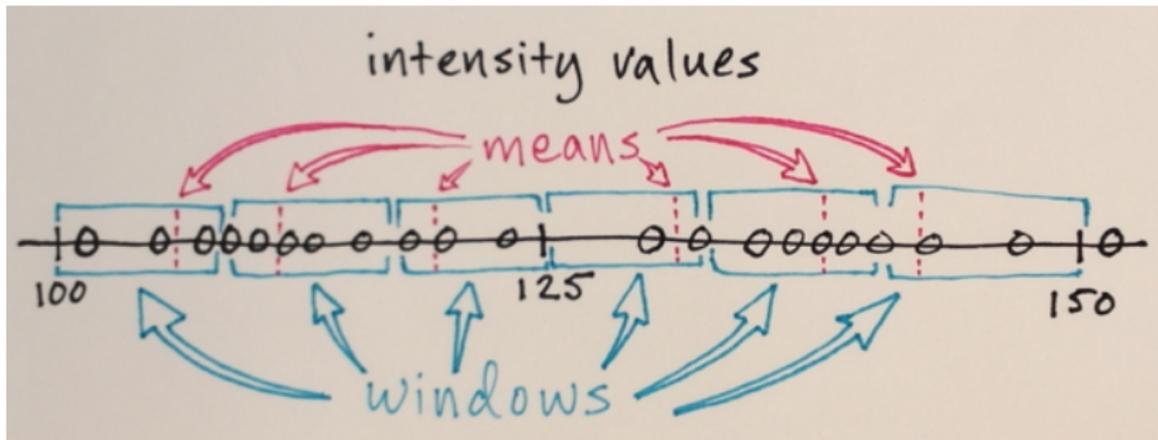


Mean-Shift : illustration du principe



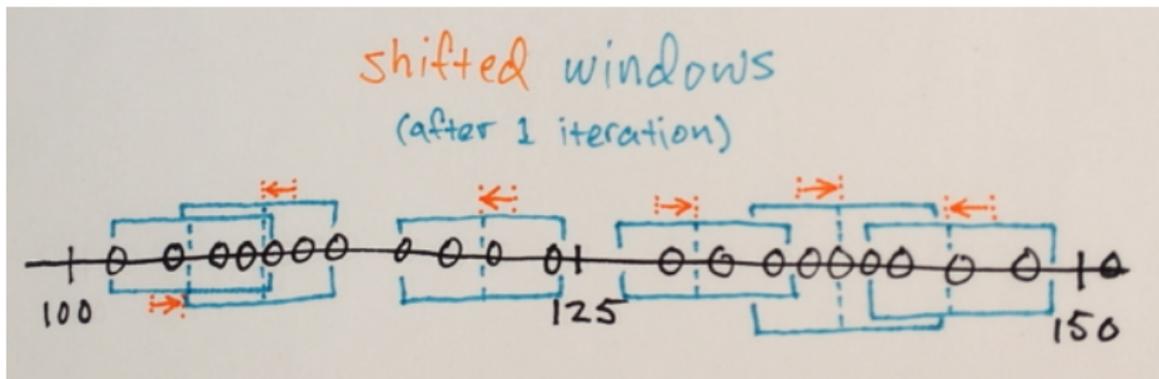


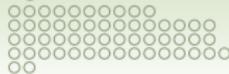
Mean-Shift : illustration du principe



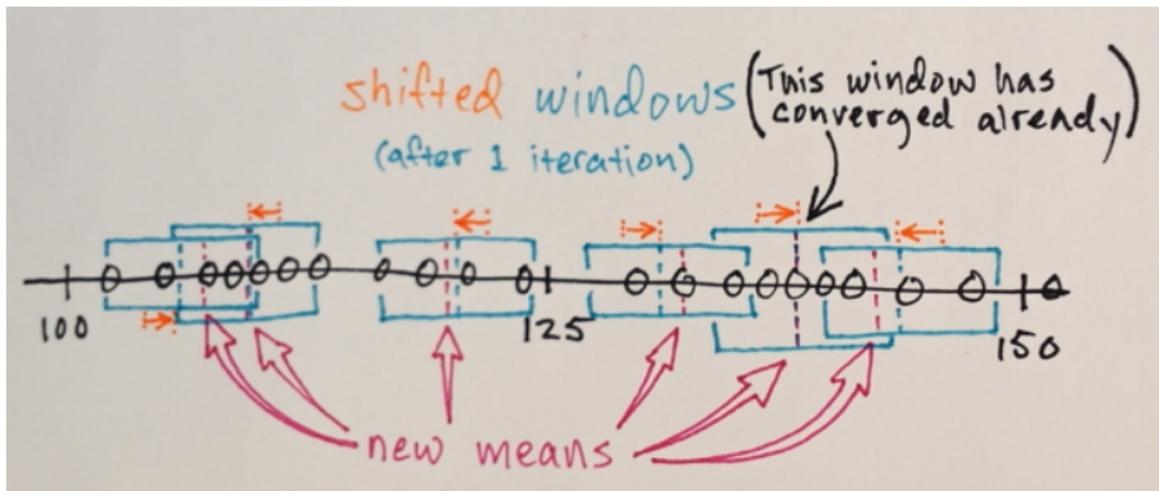


Mean-Shift : illustration du principe



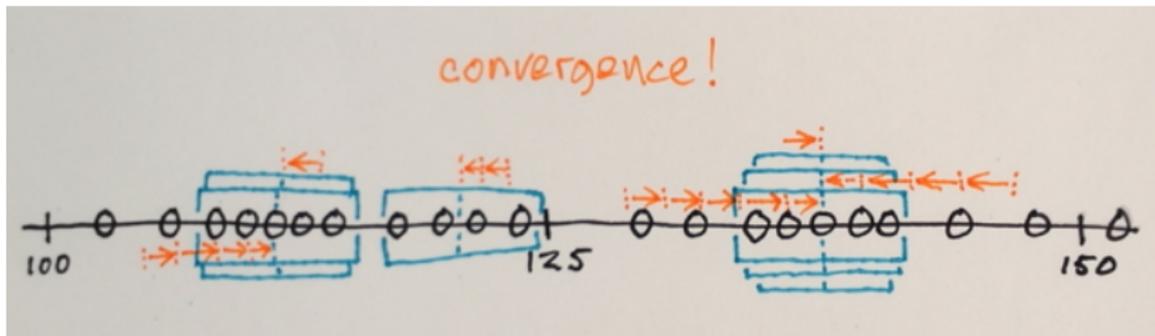


Mean-Shift : illustration du principe



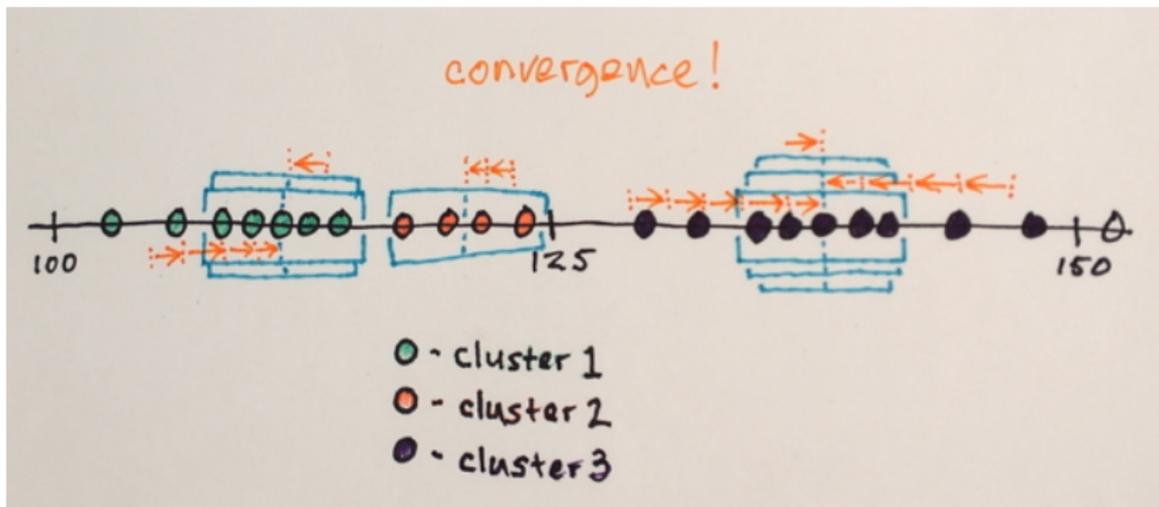


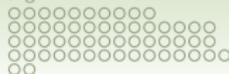
Mean-Shift : illustration du principe





Mean-Shift : illustration du principe





Mean-Shift

- **Non-paramétrique** : pas d'hypothèse sur le modèle des données
- **Solution** : analyse locale par **fenêtre de Parzen**
- Soit un échantillon de n observations $\mathbf{x}_i, i = 1 \dots n$
- Estimation de la fonction de densité $\hat{f}(\mathbf{x}), \mathbf{x} \in \mathbb{R}^d$

$$\hat{f}(\mathbf{x}) = \frac{1}{n h^d} \sum_{i=1}^n K\left(\frac{\mathbf{x} - \mathbf{x}_i}{h}\right) \quad (4)$$

où :

- Propriété du **noyau** K : positif, borné, d'intégrale égale à 1 et à **décroissance rapide**
Ici, **noyau** = fonction à appliquer sur la distance entre la donnée étudiée et celle à laquelle on se compare dans le voisinage étudié (**voisinage** ou **fenêtre de Parzen**)
- Paramètre $h > 0$ = taille du voisinage pris en compte

Ce qui implique : au lieu de fixer *a priori* le nombre de régions, on fixe un paramètre dont la valeur influe directement sur ce nombre

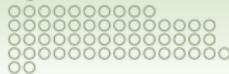


Mean-Shift

- **Choix du noyau** : Epanechnikov (noyau parabolique)

$$K\left(\frac{\mathbf{x} - \mathbf{x}_i}{h}\right) = \begin{cases} \frac{d+2}{2c_d} \left(1 - \frac{\|\mathbf{x} - \mathbf{x}_i\|^2}{h^2}\right) & \text{si } \|\mathbf{x} - \mathbf{x}_i\| < h \\ 0 & \text{sinon} \end{cases} \quad (5)$$

où c_d = volume de la **sphère unité en dimension d** et $\|\cdot\|$ = distance euclidienne



Mean-Shift

- **Avantage du noyau d'Epanechnikov** : estimation du gradient de la densité devient

$$\begin{aligned} \widehat{\nabla} f(\mathbf{x}) &= \frac{1}{n h^d} \frac{d+2}{c_d h^2} \sum_{\mathbf{x}_i \in S_h(\mathbf{x})} (\mathbf{x}_i - \mathbf{x}) \\ &= \frac{1}{n h^d} \frac{d+2}{c_d h^2} n_{\mathbf{x}} \left[-\mathbf{x} + \frac{1}{n_{\mathbf{x}}} \sum_{\mathbf{x}_i \in S_h(\mathbf{x})} \mathbf{x}_i \right] \end{aligned} \quad (6)$$

où $S_h(\mathbf{x}) =$ l'ensemble des \mathbf{x}_i se trouvant dans la sphère (en dimension d) de rayon h , centrée en \mathbf{x} , et $n_{\mathbf{x}} = \text{card} \{S_h(\mathbf{x})\}$

Si \mathbf{x} ne contient que la position, la sphère contient le voisinage en position

Si \mathbf{x} contient également la couleur, alors, la couleur sera prise en compte



Mean-Shift

- **Pour adapter à la couleur** : on remplace à chaque itération chaque \mathbf{x} par $\mathbf{x} + \delta_{\mathbf{x}} \widehat{\nabla} f(\mathbf{x})$, où $\delta_{\mathbf{x}} \widehat{\nabla} f(\mathbf{x})$ constitue un “petit déplacement” dans la direction du gradient
- En choisissant $\delta_{\mathbf{x}} = \left[\frac{1}{n h^d} \frac{d+2}{c_d h^2} n_{\mathbf{x}} \right]^{-1}$, cela revient à remplacer \mathbf{x} par la moyenne $M_h(\mathbf{x})$:

$$M_h(\mathbf{x}) = \frac{1}{n_{\mathbf{x}}} \sum_{\mathbf{x}_i \in S_h(\mathbf{x})} \mathbf{x}_i \quad (7)$$

- D'où *mean-shift* qui signifie à peu près “remplacement par la moyenne”



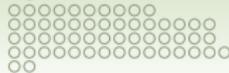
Mean-Shift : étapes

Algorithme d'estimation des modes (*mean-shift*)

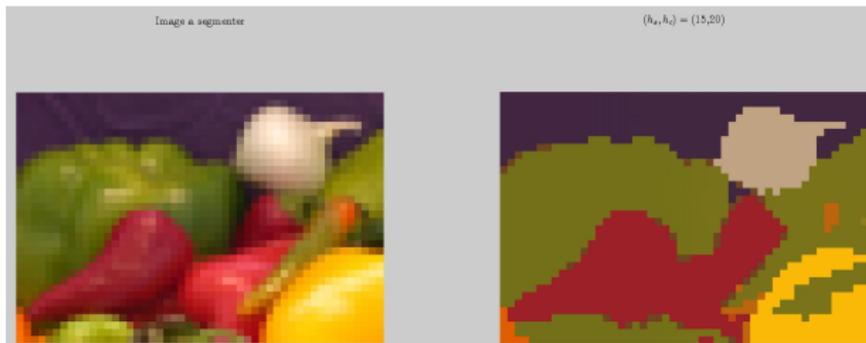
$k \leftarrow 1$

Tant qu'il existe \mathbf{x} tel que $\|\mathbf{x} - M_h(\mathbf{x})\| > \epsilon$ et que $k \leq k_{\max}$
:

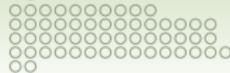
1. Pour chaque \mathbf{x} :
 - 1.1 Calcul de la moyenne $M_h(\mathbf{x})$
 - 1.2 $\mathbf{x} \leftarrow M_h(\mathbf{x})$
2. $k \leftarrow k + 1$



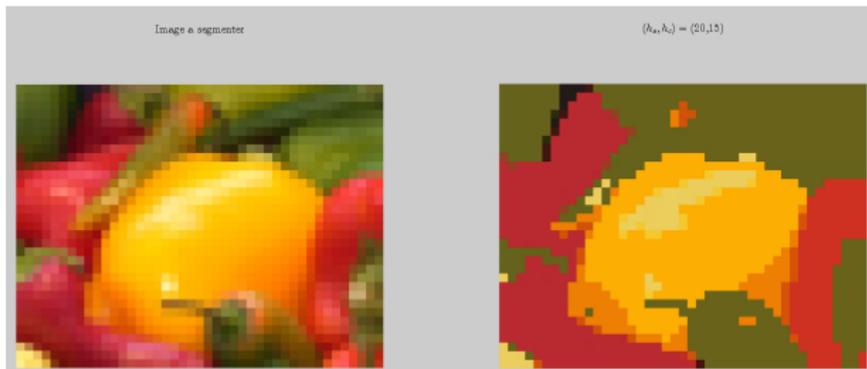
Mean-Shift : illustrations



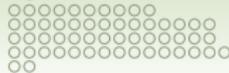
25 itérations



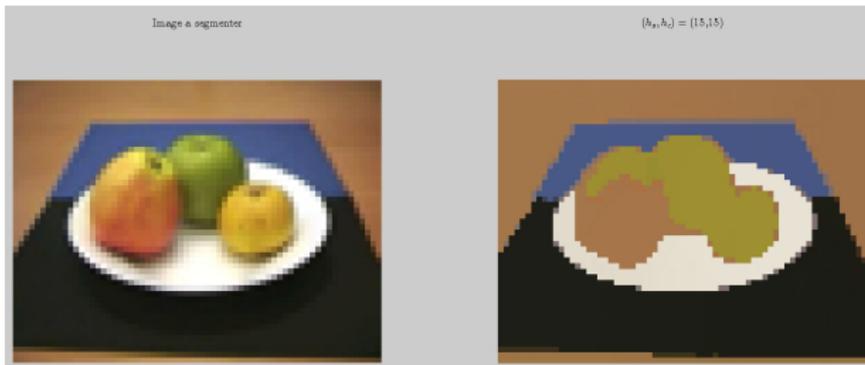
Mean-Shift : illustrations



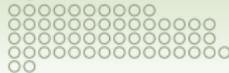
20 itérations



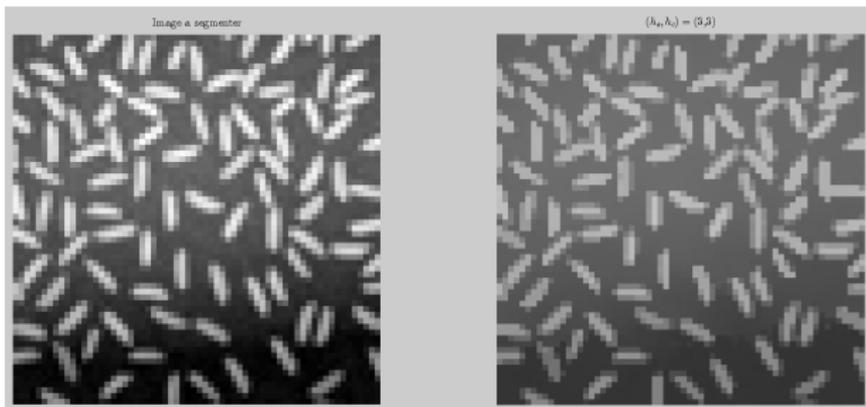
Mean-Shift : illustrations



23 itérations



Mean-Shift : illustrations



13 itérations



Segmentation par SVM

- Notion de marge maximale : Séparateurs à vaste marge ou machines à vecteur de support, *Support Vector Machine*
- Recherche de l'hyperplan qui maximisera l'écart entre les données
- **Difficultés** : que se passe-t-il quand la séparation n'est pas linéaire ?
 - Il existe une dimension où les données sont séparables
 - Notion de fonction noyau, *kernel* : noyaux polynomiaux, à fonctions de base radiales gaussiennes, sigmoïdes



Plan de la présentation

Segmentation

Superpixels



Conclusion sur la segmentation

- **Approches étudiées**

- Segmentation par contours, contours actifs
- Segmentation par histogramme, fusion/division, croissance de régions, ligne de partage des eaux
- *k*-moyenne, *mean-shift*

- Avantages/Inconvénients des différentes méthodes ?

Paramètre, **temps d'exécution**, initialisation

- Les plus utilisées généralement ?
- Complexité des algorithmes ? Applicable à de la vidéo ?
- Pourquoi utiliser la segmentation supervisée ou semi-supervisée si on peut faire de la segmentation non supervisée (avec mean shift par exemple) ?



Plan de la présentation

Segmentation

Introduction

Approches par contours

Approches par régions

Approches par classification

Conclusion

Superpixels



Introduction

- Élément de base : pixel
- Élément structuré : points/régions d'intérêt
- Primitive intermédiaire : *patches*
- Primitive intermédiaire plus évoluée
superpixels = sur-segmentation de l'image
- Quelle est la différence entre segmentation et superpixels ?



Introduction : comparaison

- Information différente de celle fournie par une segmentation
- Élément **moins discriminant qu'un objet ou un point d'intérêt**

Mais plus riche en information photométriques et/ou géométriques

- **Moins stable en terme de suivi**

Mais avec une forme permettant une meilleure adaptation à la réalité de la scène



Introduction : utilisation

- Étape de pré-segmentation ou de sur-segmentation avant une **segmentation**
Fusion/Correction en utilisant la sémantique
- **Détection et reconnaissance d'objets**
Modèle de corps humain : réduction de l'espace de recherche, meilleure précision (comparée à une boîte englobante), contrainte de spatialisation
- Le **suivi** d'objets est un problème difficile à résoudre car le fait que l'objet ou que la caméra se déplace d'une image à l'autre peut induire de nombreuses complications comme le changement
- Construction de **carte de saillance**
ajouter aux études psycho-visuelles et aux dispositifs d'*eye-tracking* pour apporter la structure



Plan de la présentation

Segmentation

Introduction

Approches par contours

Approches par régions

Approches par classification

Conclusion

Superpixels



Définition

*Un **superpixel** est une structure locale et cohérente qui représente une zone ou une partie d'un même objet*



Définition

*Un **superpixel** est une structure locale et cohérente qui représente une zone ou une partie d'un même objet*

- **Différentes définitions** : dues à l'interprétation mathématique que l'on peut faire des termes employés dans cette définition
 - Structure
 - Locale
 - Cohérente



Définition

*Un **superpixel** est une structure locale et cohérente qui représente une zone ou une partie d'un même objet*

- **Différentes définitions** : dues à l'interprétation mathématique que l'on peut faire des termes employés dans cette définition
 - Structure
 - Locale
 - Cohérente
- **Différents constructeurs** : entre 2000 et 2013 influencé par les travaux de Comaniciu et Meer



Propriétés

- Indispensable pour comprendre la mise en œuvre des constructeurs
- **Trois catégories de propriétés**
 - Apparence
 - Spatiale
 - Échelle



Propriétés d'apparence

- **Statistiques sur la colorimétrie ou l'intensité** : moyenne, écart-type . . . , des couleurs ou niveaux de gris



Propriétés d'apparence

- **Statistiques sur la colorimétrie ou l'intensité** : moyenne, écart-type . . . , des couleurs ou niveaux de gris
- **Rapport intra/inter région** : Minimisation de la **similarité entre deux régions** + maximisation de la similarité entre pixels dans une même région

$$C_{DIIR}(\mathcal{S}) = \frac{1}{\text{card}(\mathcal{S})} \sum_{S_i \in \mathcal{S}} \frac{\frac{1}{\text{card}(\mathcal{V}_{S_i})} \sum_{S_j \in \mathcal{V}_{S_i}} \sigma_{S_i \cup S_j}}{\sigma_{S_i}} \quad (8)$$



Propriétés d'apparence

- **Statistiques sur la colorimétrie ou l'intensité** : moyenne, écart-type . . . , des couleurs ou niveaux de gris
- **Rapport intra/inter région** : Minimisation de la **similarité entre deux régions** + maximisation de la similarité entre pixels dans une même région

$$C_{DIIR}(\mathcal{S}) = \frac{1}{\text{card}(\mathcal{S})} \sum_{S_i \in \mathcal{S}} \frac{\frac{1}{\text{card}(\mathcal{V}_{S_i})} \sum_{S_j \in \mathcal{V}_{S_i}} \sigma_{S_i \cup S_j}}{\sigma_{S_i}} \quad (8)$$

- **Texture** : peu exploitée



Propriétés spatiales

- **Position** : Distance



Propriétés spatiales

- **Position** : Distance
- **Connexité** : Un superpixel \mathcal{S} est connexe si et seulement si pour toutes paires de pixels $(p_i, p_j) \in \mathcal{S}$, on peut trouver un chemin reliant les deux passant uniquement pas des pixels de \mathcal{S}



Propriétés spatiales

- **Position** : Distance
- **Connexité** : Un superpixel \mathcal{S} est connexe si et seulement si pour toutes paires de pixels $(p_i, p_j) \in \mathcal{S}$, on peut trouver un chemin reliant les deux passant uniquement pas des pixels de \mathcal{S}
- **Compacité** : Rapport entre le carré du périmètre d'un superpixel et son aire

$$C_{Comp}(\mathcal{S}) = \frac{1}{\text{card}(\mathcal{S})} \sum_{S_i \in \mathcal{S}} \frac{P_{S_i}^2}{A_{S_i}} \quad (9)$$



Propriétés spatiales

- **Position** : Distance
- **Connexité** : Un superpixel \mathcal{S} est connexe si et seulement si pour toutes paires de pixels $(p_i, p_j) \in \mathcal{S}$, on peut trouver un chemin reliant les deux passant uniquement pas des pixels de \mathcal{S}
- **Compacité** : Rapport entre le carré du périmètre d'un superpixel et son aire

$$C_{Comp}(\mathcal{S}) = \frac{1}{\text{card}(\mathcal{S})} \sum_{S_i \in \mathcal{S}} \frac{P_{S_i}^2}{A_{S_i}} \quad (9)$$

- **Régularité topologique** : Chaque superpixel possède toujours le même nombre de voisins (sauf dans le cas particulier des bords)

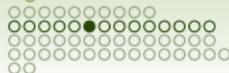


Propriétés spatiales

- **Position** : Distance
- **Connexité** : Un superpixel \mathcal{S} est connexe si et seulement si pour toutes paires de pixels $(p_i, p_j) \in \mathcal{S}$, on peut trouver un chemin reliant les deux passant uniquement pas des pixels de \mathcal{S}
- **Compacité** : Rapport entre le carré du périmètre d'un superpixel et son aire

$$C_{Comp}(\mathcal{S}) = \frac{1}{\text{card}(\mathcal{S})} \sum_{S_i \in \mathcal{S}} \frac{P_{S_i}^2}{A_{S_i}} \quad (9)$$

- **Régularité topologique** : Chaque superpixel possède toujours le même nombre de voisins (sauf dans le cas particulier des bords)
- **Contours de l'image** : détecter les contours et faire en sorte que les frontières des superpixels suivent au mieux ces contours



Propriétés d'échelle

- Support spatial : zone considérée pour mettre à jour les superpixels
- Résolution de l'images
- Nombre de superpixels souhaité



Construction

Trois familles d'approches

1. Approches s'appuyant sur la théorie des graphes
2. Approches statistiques
3. Approches par croissance de germes



Approches s'appuyant sur la théorie des graphes

- Conservation d'une certaine structure (parfois régulière)
- Nœuds = superpixels
- Liens = relations de voisinage entre superpixels
- Méthode d'optimisation = coupure de graphe



Plan de la présentation

Segmentation

Introduction

Approches par contours

Approches par régions

Approches par classification

Conclusion

Superpixels



Construction

Approches s'appuyant sur la théorie des graphes

Structure régulière

- Départ d'une grille régulière en fonction du nombre de superpixels souhaités
- Évolution de chacune des droites en courbes contraintes dans une bande autour de la droite
- Critère : Maximisation de la somme des gradients des pixels situés sur la courbe
- Avantage : Plus de superpixels dans les zones de l'image où la densité de contours est plus élevée



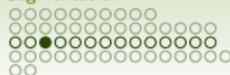
Construction

Approches s'appuyant sur la théorie des graphes

Structure régulière

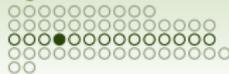
- Départ d'une grille régulière en fonction du nombre de superpixels souhaités
- Évolution de chacune des droites en courbes contraintes dans une bande autour de la droite
- Critère : Maximisation de la somme des gradients des pixels situés sur la courbe
- Avantage : Plus de superpixels dans les zones de l'image où la densité de contours est plus élevée

Structure irrégulière Un seul cas, l'approche de Felzenszwalb avec le critère Différence intra/inter région



Approches statistiques

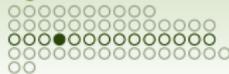
- **Inconvénient de la coupure de graphe** : coûteux en temps de calculs
- **Autre solution** : Utilisation des k -moyenne où le nombre de superpixels est donné par l'algorithme



Approches statistiques

Algorithme s'appuyant sur les k-moyennes

- **Initialisation** : distribution de manière régulière des centres C_k des superpixels
- Amélioration** : centres ajustés localement (dans un voisinage 3×3)



Approches statistiques

Algorithme s'appuyant sur les k-moyennes

- **Initialisation** : distribution de manière régulière des centres C_k des superpixels
- **Amélioration** : centres ajustés localement (dans un voisinage 3×3)
- **Itération** : k -moyenne avec ce critère

$$D_s(p, C_k) = \sqrt{d_a(p, C_k)^2 + \left(\frac{m}{S}\right)^2 d_p(p, C_k)^2}$$

où $d_a, d_p =$ distances euclidiennes entre C_k et p

S correspond au voisinage $2S \times 2S$

m permet de régler l'influence de chacune des distances



Approches par croissance de germes

- Les plus connus : les turbopixels [Levinshtein 09]
- Algorithme itératif également



Approches par croissance de germes

Algorithme des turbopixels

- Évolution de l'état Ψ^{n+1} (**le label, la classe**, à l'instant $n + 1$, des turbopixels en respectant cette formulation

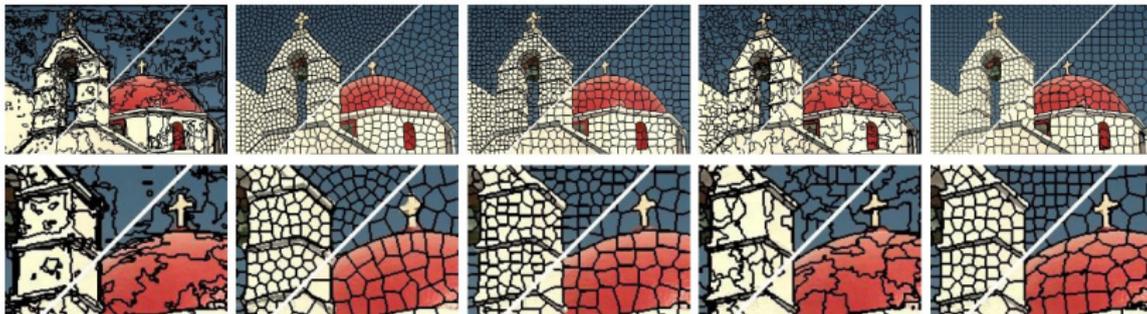
$$\Psi^{n+1} = \Psi^n - S_I S_B \|\nabla \Psi^n\| \Delta t. \quad (10)$$

- où S_I et S_B = vitesses pour accélérer ou freiner l'expansion des superpixels en fonction des contraintes extérieures
- **Contraintes : choix de S_I et S_B**
 - S_I : Courbure
 - S_B :
 - Position de la frontière des superpixels par rapport aux contours
 - Position d'un superpixel par rapport à un autre (pour empêcher deux superpixels de se recouvrir)



Comparaison de constructeur de superpixels (1)

Comment expliquer les différences observées ?



Felzenwalb

Mori

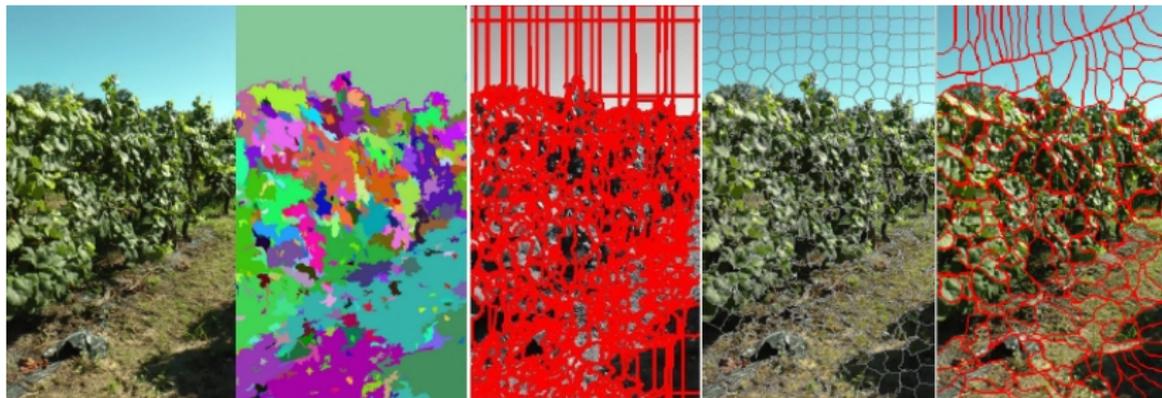
Levinshtein

Fulkerson

SLIC



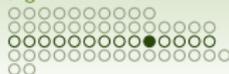
Comparaison de constructeur de superpixels (2)





Classement des constructeurs de superpixels

CAT.	RÉF.	APPARENCE			SPATIALES				
		<i>Int.</i>	<i>Reg.</i>	<i>Text.</i>	<i>Pos.</i>	<i>Comp.</i>	<i>Con.</i>	<i>Rég.</i>	<i>Cont.</i>
Graphe	[Shi 00]	–			–				
	[Felzenszwalb 04]	–	–		–				
	[moore 09]	–					–	–	–
Stat	[Achanta 12]	–		–	–	–	–		–
Germes	[levinshtein 09]	–				–	–	–	–



Conclusion sur les superpixels

Existe-t-il un constructeur de superpixel à privilégier ?

- Cela dépend du contexte applicatif (l'image à étudier, le temps et la mémoire disponible).
- Cela dépend de l'utilisation de ce résultat. S'il est primordiale d'avoir des temps de calculs faible, il est important de privilégier l'approche SLIC. Si les superpixels doivent être réguliers, il faut privilégier une approche comme SLIC ou Mori. Au contraire si nous savons que la scène est suffisamment complexe pour ne pas avoir des superpixels réguliers, il vaut mieux utiliser Felzenszwalb.



Questions d'ouverture en 2021

- Le choix du constructeur de superpixels a-t-il une influence sur la segmentation "finale" ?



Questions d'ouverture en 2021

- Le choix du constructeur de superpixels a-t-il une influence sur la segmentation "finale" ?
- Sur quel type d'image est-il intéressant d'utiliser les superpixels (image avec peu / beaucoup d'objets...)?



Questions d'ouverture en 2021

- Le choix du constructeur de superpixels a-t-il une influence sur la segmentation "finale" ?
- Sur quel type d'image est-il intéressant d'utiliser les superpixels (image avec peu / beaucoup d'objets...) ?
- Est-il possible d'utiliser la ligne de partage des eaux sur des images en couleur ? Et si oui est-elle efficace ?



Questions d'ouverture en 2022 (1/2)

- La transformation d'une image en niveaux de gris permet-elle d'obtenir des résultats suffisamment exploitables par rapport au gain de temps d'exécution obtenu ?



Questions d'ouverture en 2022 (1/2)

- La transformation d'une image en niveaux de gris permet-elle d'obtenir des résultats suffisamment exploitables par rapport au gain de temps d'exécution obtenu ?
- Pour la méthode des contours actifs, si l'objet dont on veut le contour présente une forte concavité, le contour peut-il tout de même être exact ? Est-il possible de changer dynamiquement les coefficients de flexibilité de d'élasticité, pour que le contour s'adapte plus facilement à différents types d'objets ?



Questions d'ouverture en 2022 (1/2)

- La transformation d'une image en niveaux de gris permet-elle d'obtenir des résultats suffisamment exploitables par rapport au gain de temps d'exécution obtenu ?
- Pour la méthode des contours actifs, si l'objet dont on veut le contour présente une forte concavité, le contour peut-il tout de même être exact ? Est-il possible de changer dynamiquement les coefficients de flexibilité de d'élasticité, pour que le contour s'adapte plus facilement à différents types d'objets ?
- L'algorithme Fusion/Division présentée dans les techniques de segmentations peut il être utilisé comme technique de construction de superpixels ?



Questions d'ouverture en 2022 (2/2)

- Nous avons vu dans le cours une quantité de méthodes utiles au traitement de l'images (filtres, détection de contours, segmentation...) et nous avons vu qu'il serait possible de les superposer. Comment optimiser cette superposition des différentes techniques ? Y'en a-t-il une technique à appliquer avant les autres ou pour reformuler la question, quelles sont les premières erreurs à traiter ?



Questions d'ouverture en 2022 (2/2)

- Nous avons vu dans le cours une quantité de méthodes utiles au traitement de l'images (filtres, détection de contours, segmentation...) et nous avons vu qu'il serait possible de les superposer. Comment optimiser cette superposition des différentes techniques ? Y'en a-t-il une technique à appliquer avant les autres ou pour reformuler la question, quelles sont les premières erreurs à traiter ?
- DBSCAN était une bonne alternative à k-means pour des clusters complexes. Qu'en est-il de l'utilisation de cet algorithme en traitement d'images ?



Questions d'ouverture en 2022 (2/2)

- Nous avons vu dans le cours une quantité de méthodes utiles au traitement de l'images (filtres, détection de contours, segmentation...) et nous avons vu qu'il serait possible de les superposer. Comment optimiser cette superposition des différentes techniques ? Y'en a-t-il une technique à appliquer avant les autres ou pour reformuler la question, quelles sont les premières erreurs à traiter ?
- DBSCAN était une bonne alternative à k-means pour des clusters complexes. Qu'en est-il de l'utilisation de cet algorithme en traitement d'images ?
- Comment pouvons nous évaluer la sur-segmentation ? Pouvons-nous encore utiliser les méthodes vu pour la segmentation ?