



Rapport TP3 Ima201

TP3 : Segmentation des images

Zakaria Akil

15 octobre 2023
Palaiseau, France

1 Détection de contours

1.1 Filtre de gradient local par masque

Rappelez l'intérêt du filtre de Sobel par rapport au filtre de différence, qui calcule une dérivée en utilisant simplement la différence entre deux pixels voisins.

Le filtre de **sobel** est un détecteur de contours qui permet de savoir les variations brusques de luminosité avec la connaissance de la direction de ce changement en calculant le gradient d'intensité dans chaque pixel. Par contre le filtre qui calcule la différence entre 2 pixels voisins ne détecte optimalement que les contours horizontaux/et ou verticaux (4 voisins).

Est-il nécessaire de filtrer l'image avec un filtre passe-bas avant d'utiliser le filtre de Sobel ?

Le filtre de sobel comprend déjà un pré filtrage passe bas qui enlève les très hautes fréquences et qui ne correspondent pas généralement à des bords mais d'un bruit indépendant.

Le seuillage de la norme du gradient permet d'obtenir des contours. Commentez la qualité des contours obtenus (robustesse au bruit, continuité, épaisseur, position...) lorsque l'on fait varier ce seuil.

Les contours sont de plus haute robustesse quand le seuil est petit. Plus le seuil est grand, plus on perd la sensibilisation aux contours ce qui est normal. Par contre la détection de quelques bruits est toujours maintenue même si le seuil est assez grand ce qui montre leur fréquence élevée.

seuil grand \rightarrow dégradation/ perte de la continuité des contours

1.2 Maximum du gradient filtré dans la direction du gradient

Quel critère de qualité est optimisé par ce procédé ?

Cette technique évite d'avoir dans l' image résultante toute une zone colorée juste parce que les gradients sont supérieurs aux seuils, cela évite de considérer les contours dans tous les sens et donne des contours minces avec plus de sélectivité. Ainsi le résultat est sous forme de contours significatifs. C' est plus robuste aux bruits.

Les critères de détection de Canny

Les 3 critères de détection de Canny sont les suivants :

- bonne détection : faible taux d'erreur dans la signalisation des contours,
- bonne localisation : minimisation des distances entre les contours détectés et les contours réels,
- clarté de la réponse : une seule réponse par contour et pas de faux positifs

1.3 Filtre récursif de Deriche

Le temps de calcul dépend-il de la valeur de α ? Expliquez pourquoi.

Oui alpha change le temps de calcul, $dt(\alpha = 0.4) = 5s$ alors que $dt(\alpha = 2) = 8s$. Cela est normal car il s'agit d'un algorithme récursif.

1.4 Passage par zéro du laplacien

2 Seuillage avec hystérésis

Quand on augmente le rayon, on augmente l'intensité des lignes plus fines - on perçoit plus de lignes, et avec une intensité plus forte.

- L : 1/ H : 10 : Certains contours ne sont pas fermés
- L : 5/ H : 10 : Des valeurs trop proches donnent des observations moins précises/pertinentes, il faut donc garder un bon écart entre les 2 seuils
- L : 5/ H : 15 : on a augmenté les 2 seuils, il reste des contours non fermés
- L : 1/ H : 5 : On obtient des résultats très satisfaisants, les contours sont fermés sans qu'il reste trop de bruit

3 Segmentation par classification : k-moyennes

image niveaux de gris

- En utilisant 2 classes, on sacrifie l'information relative aux noyaux noirs, mais on réussit à bien représenter les cellules et les noyaux clairs.

- En augmentant à 3 classes, on comble le manque d'information dans les noyaux noirs.

- Lorsque le paramètre "random-state" est défini sur "None," l'initialisation est aléatoire, ce qui garantit que les classes restent stables dans tous les cas.

- En ce qui concerne l'image "muscle.tif," l'utilisation de seulement 2 classes est insuffisante, car cela entraîne la perte des cellules noires. Si le nombre de classes est trop bas, la variation des couleurs à l'intérieur d'une cellule peut conduire à sa segmentation en plusieurs classes.

- En utilisant le filtre médian, la texture des cellules est perdue, ce qui rend les cellules homogènes (avec une seule couleur). Dans ce cas, il n'est pas nécessaire d'avoir un grand nombre de classes, car le problème précédent a été résolu.

image avec couleurs

On observe que l'application de ce procédé entraîne des couleurs plus pâles, ce qui entraîne la perte de la texture et des dégradés des pétales, ainsi que la suppression de certains éléments.

- 'on utilise 5 classes, seules les couleurs jaune, blanc, et marron/rouge sont conservées.

- Avec 7 classes, on récupère la couleur rouge, mais certaines formes sont encore manquantes, notamment les pétales violets foncés en bas à droite.

- En passant à 15 classes, on obtient une meilleure représentation, mais il manque toujours certaines formes.

- Finalement, avec 20 classes, on parvient à saisir les détails des différentes couleurs présentes dans une même pétale.

- La solution consiste à augmenter progressivement le nombre de classes de couleurs (n_{colors}) jusqu'à retrouver l'image originale de "carte.tif".

4 Seuillage automatique : "Otsu"

- On cherche à optimiser selon le critère suivant : obtenir la plus grande différence d'intensité moyenne entre les classes, de manière à obtenir la meilleure répartition possible des pixels entre les deux classes. Cela équivaut à minimiser la dispersion intraclassique, comme mentionné dans l'énoncé.

- Sur l'image "cell.tif", il n'y a pas d'amélioration notable par rapport à la méthode K-Means.

- Quant à la méthode Otsu, elle ne fonctionne pas avec l'image "dice.tif" car elle nécessiterait au moins trois classes.

- En ce qui concerne l'image "pyra.tif," cette méthode est utilisable si l'on segmente l'histogramme en plusieurs sous-histogrammes, car la méthode Otsu permet une séparation nette et précise.

- Pour l'image "lena.tif," les contours sont bien restitués et bien distincts grâce à cette méthode.

5 Croissance de régions

- La différence d'intensité moyenne entre le voisinage d'un pixel et la zone de départ doit être inférieure à $threshold * S_0$, ce dernier étant le critère de sélectivité, où une valeur plus élevée de $thresh$ rend la sélection moins stricte ($thresh > 1$ car les intensités des pixels sont des valeurs entières).

- L'initialisation en (X_o, Y_o) doit correspondre à un point de matière blanche. En passant de threshold 2 (Fig. 3) à threshold 5 (Fig. 4), des résultats considérablement améliorés sont obtenus. Avec un rayon de 6, des anomalies en forme de pâtés apparaissent. En revanche, avec un rayon de 3 (Fig. 5), l'algorithme est plus lent, mais la forme de la zone sélectionnée est plus précise, et le résultat est excellent.

- Augmenter le threshold peut conduire à accepter la matière grise, sans la distinguer de la matière blanche.

- Le prédicat ne dépend pas de l'évolution de la région, mais uniquement du voisinage autour du pixel examiné et du voisinage du pixel de départ. Il compare la moyenne d'intensité du voisinage du pixel examiné à la moyenne d'intensité du voisinage du "centre" (pixel de départ), qui reste invariable. Il pourrait être calculé pour tous les pixels simultanément.

- On élimine l'utilisation du périmètre et on calcule directement la différence de moyenne d'intensité du voisinage de chaque pixel de l'image par rapport à la moyenne d'intensité du voisinage du "centre" (pixel de départ). Si le prédicat est confirmé, le pixel est ajouté à la zone de sélection.

- Il serait nécessaire de remplacer la moyenne d'intensité du voisinage du "centre" (pixel de départ) dans le prédicat par la moyenne d'intensité sur toute la région sélectionnée à chaque étape.