Segmentation d'une image par fusion de régions

Sujet proposé par Philippe Chassignet

Philippe.Chassignet@polytechnique.fr

1 Introduction

La segmentation d'une image est l'opération qui conduit à distinguer différentes régions dans cette image. On s'intéresse ici à une méthode qui construit explicitement des régions, plutôt que de caractériser les frontières entre régions.

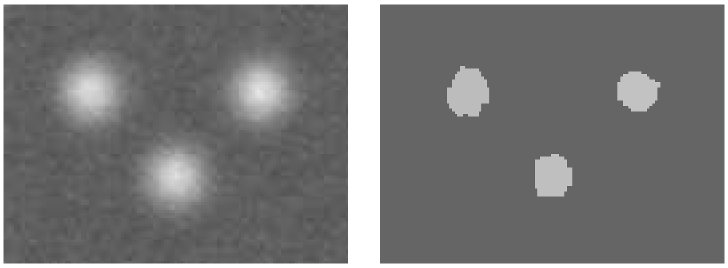

Figure 1: Exemple de segmentation en 4 régions (dont le fond)

La méthode proposée consiste à construire un état initial où chaque pixel de l'image constitue une région élémentaire, puis à fusionner progressivement les régions connexes qui satisfont un certain critère d'homogénéïté. Comme exemple de critère simple, on peut considérer l'écart entre les valeurs moyennes des pixels dans deux régions. Si cet écart est inférieur à un certain seuil, on peut fusionner les deux régions en une seule.

L'ordre dans lequel sont réalisées les fusions influe naturellement sur le résultat. Il convient de commencer par fusionner les régions les plus "proches" selon le critère considéré.

Le résultat est une image où tous les pixels d'une région sont marqués de la même valeur. Le plus naturel est de prendre pour chaque région la valeur moyenne de ses pixels dans l'image d'origine. On pourra aussi produire un coloriage selon le numéro de la région.

2 Détail du sujet

Les difficultés de ce sujet résident essentiellement dans la représentation de l'état courant et dans la manière de le faire évoluer, notamment l'opération de fusion de deux régions et la gestion de l'ordre des fusions.

Les moyens pour lire, écrire et visualiser des images sont indiqués dans une page web annexe [1]. Une image sera alors vue et utilisée simplement comme un tableau de valeurs entières.

2.1 Structure d'image matricielle

Les coordonnées dans l'image s'expriment par des nombres entiers positifs. L'origine est placée dans le coin en haut et à gauche de l'image, l'axe des x est orienté vers la droite, et l'axe des y est orienté vers le bas, ce qui est conforme aux conventions des bibliothèques graphiques et des formats de fichiers usuels.

Un *pixel* (x, y), élément d'une image discrétisée sous forme matricielle, est le carré de coordonnées $[x, x+1] \times [y, y+1]$. Dans une image en niveaux de gris, chaque pixel a une valeur entière, comprise entre 0 et 255 ($0 =$ noir, $255 =$ blanc).

Pour la relation de "4-connexité", un pixel (sauf au bord de l'image) a 4 voisins directs avec chacun desquels il partage un côté.

2.2 Régions et graphe d'adjacence

Une région est un ensemble de pixels qui seront 4-connexes par construction. La seule fonction requise est de pouvoir énumérer les pixels qui composent une région. Il est utile pourtant de mémoriser, pour chaque région, le nombre de pixels, la somme de leurs valeurs (pour calculer la moyenne) et d'autres informations qui serviront à déterminer rapidement des critères de fusion. Lors de la fusion de deux régions, l'une est choisie arbitrairement pour "absorber" l'autre. Les données de la région absorbante sont alors "augmentées" de celles de la région absorbée. Il est conseillé que chaque région absorbée conserve un lien sur la région absorbante, on pensera pour cela à l'algorithme "union-find".

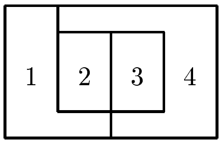

Figure 2: Exemple de segmentation en 4 régions

Deux régions sont adjacentes si au moins un pixel de l'une est voisin direct (pour la "4-connexité") d'un pixel de l'autre. Cela définit très naturellement une structure de graphe non orienté. La figure 2 donne un exemple, les adjacences sont alors $(1 \leftrightarrow 2)$, $(1 \leftrightarrow 4)$, $(2 \leftrightarrow 3)$, $(2 \leftrightarrow 4)$ et $(3 \leftrightarrow 4)$.

La mise à jour des adjacences lors d'une fusion constitue l'une des difficultés principales de ce sujet. On étudiera sur la figure 2 les conséquences d'une absorption de la région 4 par la région 1. Les adjacences restantes sont alors $(1 \leftrightarrow 2)$, $(1 \leftrightarrow 3)$ et $(2 \leftrightarrow 3)$. Le coût de mise à jour doit être linéaire dans le nombre d'adjacences concernées, c'est-à-dire, celles des deux régions à fusionner ainsi que celles de toutes leurs voisines directes. À l'occasion de cette mise en jour, on devra aussi réactualiser la priorité des adjacences, comme indiqué dans la section suivante.

On initialisera cette structure de manière à représenter le quadrillage complet de l'image, où chaque pixel constitue une région différente, avec des adjacences aux pixels voisins.

2.3 Contrôle des fusions

L'algorithme consiste donc à réaliser itérativement des fusions, en commençant par les régions les plus "proches", jusqu'à atteindre un certain critère d'arrêt (seuil de qualité, nombre de régions restantes, ...). Pour gérer l'ordre des fusions, il s'impose d'utiliser une structure de file d'attente avec priorité dans laquelle seront placées les adjacences entre régions candidates à une fusion. Toutes les adjacences initiales seront donc insérées dans une telle file, avec un coût en $O(\log N)$ par insertion. L'élément en tête de la file donnera les deux régions à fusionner en priorité.

Ensuite, à l'issue de chaque fusion, il faut réactualiser les priorités des adjacences entre la région résultante et les régions environnantes. La réactualisation de la priorité d'une adjacence se fait en $O(\log N)$ si l'on a mémorisé sa position dans la file.

Remarque : Après certaines fusions, des éléments de la file ne sont plus d'actualité car ils sont devenus redondants avec d'autres adjacences ou car ils concernent des régions déjà fusionnées. Il y a plusieurs solutions pour les écarter : les supprimer tout de suite, les envoyer en queue de file ou les détecter quand ils arrivent en tête.

3 Travail demandé

Pour ce projet, il s'agira donc :

1. d'écrire le programme de segmentation par fusion, en suivant les indications donnéee ci-dessus,
2. d'expérimenter pour différentes images et pour différents critères de fusion.

On trouvera quelques critères dans l'article [2], cf. section 3, les quatre premiers se mettent facilement en œuvre, alors que le dernier (3.5) demande plus de travail et constitue une extension. L'article [3], section V.B, donne un cadre théorique pour le choix de bons critères.

4 Extensions possibles

4.1 Gestion des frontières

Cette extension permet de ne prendre en compte que les pixels de part et d'autre d'une frontière, cf. 3.5 de [2] plutôt que tous les pixels des deux régions considérées. Cela consiste à coder le détail de chaque frontière, par exemple en énumérant les déplacements élémentaires qui la composent, chacun de ces déplacements séparant une paire de pixels et contribuant au critère.

4.2 Segmentation préalable

Les deux articles cités en référence suggèrent le recours à des prétraitements pour améliorer les temps de calcul et la qualité du résultat. Les sections III et IV de [3] constituent une base pour de telles extensions. Les élèves intéressés prendront contact avec P. Chassignet pour définir des objectifs raisonnables dans le cadre de ce projet.

References

[1] `http://www.enseignement.polytechnique.fr/profs/informatique/Philippe.Chassignet/03-04/REGIONS/index.html`

[2] T. Brox, D. Farin and P.H.N. de With, *Multi-stage region merging for image segmentation*, in *Proceedings of the 22nd Symposium on Information Theory in the Benelux.*
disponible sur le web :
`http://www.informatik.uni-mannheim.de/informatik/pi4/publications/library/Farin2001b.pdf`

[3] K. Haris, S.N. Efstratiadis, N. Maglaveras and A.K. Katsaggelos, *Hybrid image segmentation using watersheds and fast region merging*, IEEE Trans.on Image Processing, vol. 7, no. 12, pp. 1684-1699, dec. 1998.
disponible sur le web :
`http://ivpl.ece.northwestern.edu/Publications/Journals/1998/IEEETransIP98c.pdf`