
Constrained Connectivity for Hierarchical Image Partitioning and Simplification

Compte rendu d'article



Kevin TRANCHO
Master Informatique 2^{ème} année
Spécialité Sciences de l'image
Année 2018 - 2019



1 Contexte et objectifs

Aujourd'hui nous souhaitons obtenir un partitionnement d'images dans le but de répondre à des problèmes divers tels que de la simplification d'une image ou sa segmentation ceci pour satisfaire les besoins d'un utilisateur. Cependant c'est un problème difficile et aucune méthode n'est encore optimale pour répondre à ce problème quelque soit l'image à traiter, car les critères de qualité du résultat peuvent varier d'une application à une autre. Pour répondre à ce problème Soille [Soi08] propose une méthode qui se base sur des critères de connexions sur les composants de la segmentation. Sa méthode s'inspire sur l'idée de construire un 'puzzle' depuis une image en niveaux de gris ce qui laisse à un utilisateur la possibilité d'étudier différents critères statistiques sur les différentes zones par exemple. L'idée est de proposer une méthode qui se base sur de simples différences de mesures sur les différents pixels de l'image pour en conserver les différents composants sous réserve de ne pas dépasser un seuil donné ce qui fait suite à l'idée proposée par Nagao et al. [NMI79] bien plus tôt dans la littérature.

Ici la méthode de Soille [Soi08] cherche à pro-

poser une amélioration de l' α -connectivité proposée par la notion de plateaux en morphologie mathématique [SS93] [SS95] [CSS⁺97] [MM99]. L'objectif est d'améliorer le fait que la segmentation donnée par l' α -connectivité est peu précise car en effet pour que deux pixels fassent partie d'un même composant il suffit qu'il existe un chemin entre ces deux pixels dont chaque arête n'excède pas le seuil α . Le but de la méthode est aussi de proposer une hiérarchie de classes d'équivalences tout en offrant une grande possibilité et simplicité de paramétrage du résultat final souhaité pour la simplification de l'image.

2 Travail et principaux résultats

2.1 Prérequis théoriques

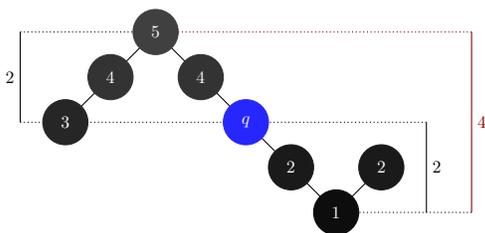
Dans son article, Soille [Soi08] donne à la fois une définition formelle des différentes notions qu'il est possible de retrouver facilement dans les sections 2 à 3 de son article, mais aussi une explication textuelle de celles-ci.

Dans un premier temps, les différentes applications supposent que nous travaillons sur une image en nuances de gris. Depuis cette image,

nous allons en extraire un graphe dont les arêtes correspondront à la différence de valeurs évaluées par $R(f(p) - f(q))$ pour p et q des nœuds voisins et f la fonction qui associe sa valeur à une position donnée (ceci sera illustré dans notre exemple en section 2.3).

Dans les prérequis théoriques pour comprendre l'article, une première notion abordée est l' α -connectivité. Dire qu'une classe d'équivalence est α -connectée veut simplement dire que chacun des éléments de cette classe est connecté à chacun des autres par un chemin qui passe par des arêtes de poids au plus α . Cette définition garantit qu'une classe $\alpha - CC(p) \subseteq \beta - CC(p)$ dès lors que $\beta \geq \alpha$ ce qui implique la structure hiérarchique des classes.

Ensuite est énoncée la notion d' (α, ω) -connectivité celle-ci impose que tous les chemins entre deux éléments de la classe soient bornés par ω de sorte d'éviter à un chemin d'atteindre une différence entre une valeur maximale et minimale qui dépasse ω . Ceci en prenant pour classes d'équivalence la plus grande classe α_i -connectée avec $\alpha_i \leq \alpha$ qui vérifie partout la condition sur ω . Ceci fait suite à la définition d'Hambrusch et al. [HHM94] qui ne garantit pas la transitivité de la relation comme illustré en figure suivante pour un point q contrairement à la définition donnée par Soille [Soi08] (définition formelle en équation (2) de l'article) :



C'est-à-dire que dans la définition donnée par Hambrusch et al. [HHM94] q serait $(1, 2)$ -connecté mais pas pour Soille [Soi08] où l'exemple demanderait d'être $(1, 4)$ -connecté.

Une autre définition pertinente énoncée est celle de l' α -indice de connexion qui équivaut au ratio des arêtes de poids inférieur à α sur la taille

du sous graphe étudié (donné par Soille [Soi08] en équation (5)). On dira que si ce ratio vaut 1, alors toutes les arêtes composant le sous-graphe sont au plus α ce qui conduit à dire que le composant est α -fortement connecté. Plus précisément les classes d'équivalences des composants α -fortement connectés sont les plus grands composants α_i -connectés avec $\alpha_i \leq \alpha$ qui sont α -fortement connectés que Soille [Soi08] donne en définition à l'équation (6) de son article.

Ces notions sont les principales notions et sont celles qui seront principalement analysées dans les résultats proposés. Bien qu'il soit aussi donné une définition supplémentaire qui est celle de l' (α, ω, β) connectivité qui impose aux classes d'équivalence d'être à la fois (α, ω) -connectées mais aussi d'avoir un α -indice de connexion au moins β . Ce qui peut être vu comme une définition qui peut combiner et/ou interpoler les définitions de l' (α, ω) -connectivité et de l' α -forte connectivité.

2.2 Méthode

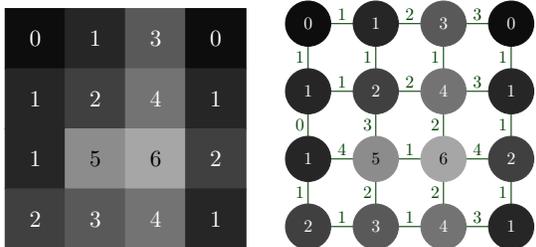
Concernant la méthode, l'algorithme proposé se base sur l'utilisation d'une file de priorité qui est utilisée pour classer les pixels candidats pour étendre la classe d'équivalence actuelle et l'utilisation d'une pile qui contiendra l'ensemble de pixels à ajouter à la classe actuelle pour l'étendre jusqu'à ce qu'à la fin du tour de boucle la classe soit une classe d'équivalence de l' (α, ω) -connectivité finale.

Il est proposé de traiter les pixels non assignés à une classe d'équivalence en assignant la différence locale à regarder de sorte de ne pas s'étendre sur une classe existante (lignes 9 à 14 de l'algorithme) et ajouter les voisins qui peuvent potentiellement être utilisés comme points de départ pour construire la segmentation à la file de priorité (lignes 15 à 18). Lorsqu'on a réussi à détecter un composant α_i -connecté dans son intégralité, on l'ajoute à la classe actuelle (lignes 26 à 33). Dans le cas où l'extension de la classe par le pixel traité ne respecte pas la condition imposée par ω : $R(\alpha_i - CC(\text{pixel})) \leq \omega$ ou que

l'ajout entraîne une fusion de zone alors on vide la pile (lignes 42 à 45) de sorte de ne pas l'étendre avec le α_i actuel qui est ici trop grand. Ensuite on regarde les voisins du pixel candidat, dans le cas où il serait valide et entraînerait une fusion alors le α_i n'est pas valide (lignes 48 à 53), sinon on l'ajoute à condition qu'il soit un candidat valide pour le α_i donné et que sa distance au pixel candidat soit plus petite que les différences de valeurs testées précédemment pour ce voisin (lignes 56 à 61).

2.3 Illustration du concept $d'(\alpha, \omega)$ -connectivité

Pour illustrer la méthode proposée par la segmentation basée sur la connectivité (α, ω) décrite précédemment, nous proposons l'exemple suivant :

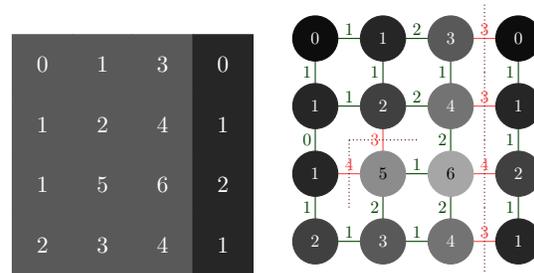


Sur lequel est représenté à droite l'image donnée en entrée et sur la gauche le graphe associé à cette image pour aider à la compréhension du mécanisme. Ici nous appliquons la méthode dans le cadre de la 4-connectivité pour le voisinage entre les pixels. À chaque pixel on associe son niveau de gris et sur le graphe représenté les arêtes sont pondérées par la différence de valeurs entre les pixels voisins. Notons que cette différence peut être par exemple la norme L^2 si ces pixels sont colorés ou tout autre mesure pertinente.

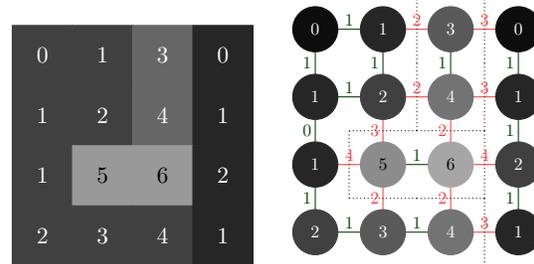
Sur les schémas suivants sont illustrés les segmentations pour la méthode qui traite de l' α -connectivité. Où chaque classe d'équivalence est colorée par la moyenne de ses valeurs. Notons que sur le graphe nous représentons en vert foncé les arêtes légales pour former un chemin d'un pixel à un autre et en rouge clair (avec séparation en pointillés) les arêtes illégales à traverser pour le

α donné.

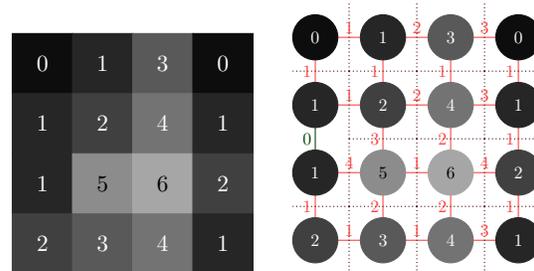
Ce qui conduit pour $\alpha = 2$ à la représentation ci-dessous où la segmentation est formée de deux classes d'équivalences.



Dans le cas de $\alpha = 1$, nous obtenons la simplification ci-dessous, c'est-à-dire obtenue depuis la segmentation de 4 classes d'équivalence.



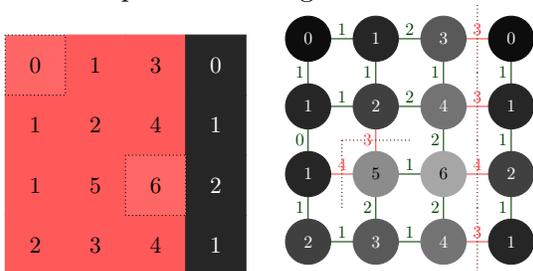
Par suite dans le cas $\alpha = 0$, nous revenons à notre image de départ comme illustré ci-dessous, car en effet les seuls pixels liés sont ceux de même valeur.



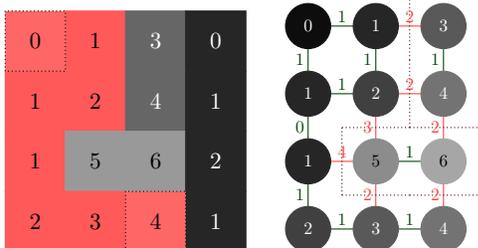
Nous souhaitons maintenant construire les classes dans le cas de la (α, ω) -connectivité. Dans l'exemple que nous donnerons, nous avons choisi de poser $\alpha = 2$ et $\omega = 2$. Pour illustrer le procédé de manière plus intuitive et comprendre graphiquement comment obtenir le résultat final, nous proposons de le visualiser comme la résolution récursive des zones α_i -connectées qui ne respectent pas la contrainte donnée par ω en résolvant les zones à problèmes en demandant les zones $(\alpha_i - 1, \omega)$ -connectées et en conservant les

classes d'équivalence au α_i donné dès lors qu'elles respectent $R(\alpha_i - CC) \leq \omega$.

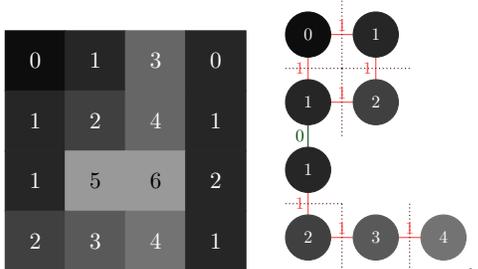
Dans l'exemple suivant, on peut remarquer que $\alpha = 2$ pose problème, c'est-à-dire que la zone de droite est à $R(\alpha_i - CC((1,1))) = 6 - 0 = 6 > \omega = 2$. Cependant la zone de gauche est à $R(\alpha_i - CC((4,1))) = 2 - 0 = 2 \leq \omega = 2$ elle est donc une classe valide pour la (α, ω) -connectivité et fait donc partie de la segmentation.



Ici, $\alpha_i = 2$ est trop grand pour que la zone de droite soit (α, ω) -connectée. Le but est de trouver les plus grandes zones z α_i -connectées telles que $R(\alpha_i - CC(z)) \leq \omega$. Ici une projection du graphe de la zone posant problème dans le graphe $\alpha_i - 1 = 1$ laisse apparaître deux classes d'équivalence (α, ω) -connectées, en effet ce sont les plus grandes α_i -connectées de la hiérarchie avec $\alpha_i \leq \alpha$ qui vérifient le critère précédent vis-à-vis d' ω .

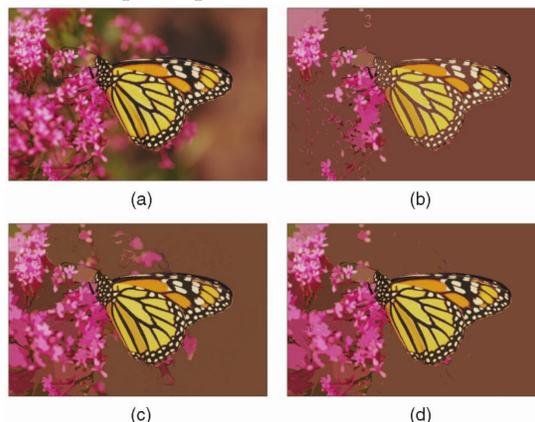


Cependant la zone tout à droite pose encore problème car $4 - 0 = 4 > \omega$. Le seul choix pour cette composante est $\alpha_i = 0$, d'où la simplification finale suivante :



2.4 Résultats

L'un des principaux résultats est le suivant :



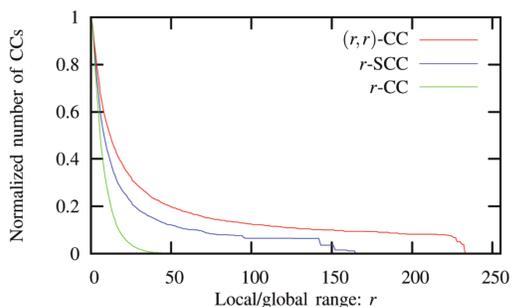
Ce qui est illustré ici est la simplification d'une image (a) où pour les résultats (b), (c) et (d) on a gardé la segmentation qui correspond à représenter un dixième du nombre de composants qu'on obtient sur une segmentation pour $\alpha = 0$. Ce qui se traduit par les segmentations suivantes :

- En (b) : simplification pour les composants 9-connectés.
- En (c) : pour les composants (150, 150)-connectés.
- En (d) : pour les composants 27-fortement connectés.

Ce que nous pouvons remarquer est que dans le cas α -connecté (b) la simplification a fait disparaître la séparation entre le papillon et le fond de l'image ce qui n'est pas souhaité lors d'une segmentation. Les résultats en (c) et (d) donnent une meilleure segmentation. Il est malgré tout possible de noter que la relation d' (α, ω) -connexion garde principalement de grandes zones, telles que par exemple des détails de fleurs en fond et estompent par exemple des légers détails blancs dans les zones jaunes sur les ailes du papillon qui sont conservées lorsque nous sommes α -fortement connectés. Autrement dit l' (α, ω) -connexion semble ici plus s'appuyer sur le fait de conserver de grandes zones connectées si elles peuvent être significative alors que des plus petits détails peuvent être gardés s'ils sont α -fortement connectés dans l'autre cas.

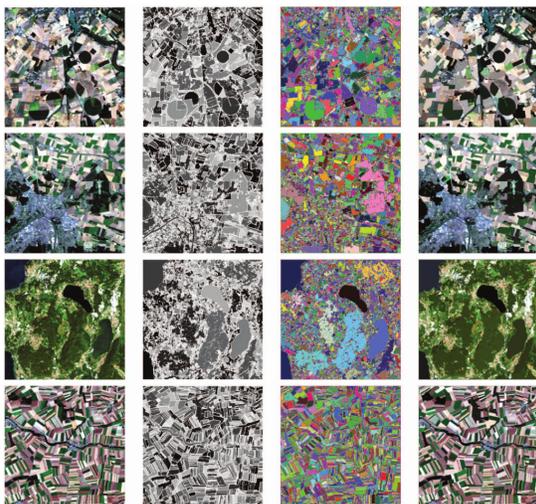
De plus l'une des autres comparaison entre les

méthodes est illustré par la figure suivante :

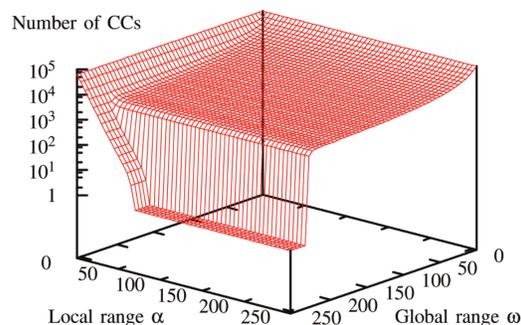


Ce qui est montré ici est la souplesse offerte par le paramétrage de l' (α, ω) -connectivité pour la simplification. C'est-à-dire qu'elle offre plus de possibilités de réglage des paramètres pour obtenir une simplification voulue que les deux autres méthodes illustrées.

Un autre résultat intéressant est par exemple illustré par la figure ci-dessous. Car en effet, alors que beaucoup de méthodes de segmentation demandent une interaction avec un utilisateur, la méthode proposée semble aussi donner de bons résultats lors de segmentations automatiques.



Un problème qui pourrait être posé est le réglage des paramètres α et ω . Ce qui est étudié par la figure suivante :



qui étudie le nombre de composants obtenus en fonction des paramètres, et il semble qu'en effet que poser $\alpha = \omega$ semble pertinent. Car en effet, lorsque α dépasse ω ceci n'a plus d'influence sur le nombre de composants de la segmentation.

3 Analyse critique

3.1 Les points faibles

Il est proposé une définition de l' (α, ω, β) -connectivité dont β laisse la possibilité de paramétrer l'indice de connexion β , mais aucun résultat n'est présenté pour montrer son influence ou son paramétrage. En particulier qu'il est montré des résultats aussi sur la segmentation par classes α -fortement connectées (ce paramètre pourrait montrer les possibilités offertes en montrant les combinaisons possible de l' (α, ω) -connectivité avec α -forte connectivité : cas $\beta = 1$).

Un autre point faible ici de la méthode peut être par exemple le temps de calcul, c'est-à-dire que ceci ne permet pas de l'utiliser dans des applications temps-réel. En effet, il est dans un ordre de grandeur de l'ordre de la seconde pour des images que l'on peut souhaiter traiter. De plus, l'algorithme et la méthode sont très dépendants des valeurs et connexions entre les différents pixels, ce qui rend le parallélisme difficile à mettre en place pour améliorer le temps de calcul.

Notons que j'ai trouvé ce qui me semble être une typographie dans l'algorithme (figure 10 de l'article [Soi08]), c'est-à-dire que pour désigner la distance entre la valeur des pixels de l'image

en deux positions, il est utilisé $R\{f(p) - f(q)\}$ à la ligne 8 et $\text{abs}(f(p) - f(q))$ à la ligne 46, la première notation semble plus générique bien que la seconde est en réalité implicite sur des niveaux de gris.

3.2 Les points forts

L'un des avantages de la méthode proposée est sa possibilité de ne pas être dépendant de la manipulation par un utilisateur tout en donnant des résultats de simplification d'une image intéressante. Notons de plus que les illustrations des résultats et tests me semblent pertinents, en effet Soille [Soi08] compare ses résultats d'autres méthodes pertinentes ici, illustre le paramétrage de sa méthode, donne les temps de calculs attendus sur les différentes tailles d'images.

Le document est de mon point de vue bien écrit, c'est-à-dire qu'il énonce le contexte, présente rapidement l'état de l'art, puis, avant de présenter les résultats de la méthode et comparaisons, donne les prérequis pour comprendre la suite de l'article ceci par des détails et illustrations pour comprendre les notions mais aussi en plus de nous donner les définitions de chaque notion ce qui rend l'article facile à lire et pédagogique.

Références

- [CSS⁺97] José Crespo, Ronald W Schafer, Jean Serra, Christophe Gratin, and Fernand Meyer. The flat zone approach : a general low-level region merging segmentation method. *Signal Processing*, 62(1) :37–60, 1997.
- [HHM94] Susanne Hambrusch, Xin He, and Russ Miller. Parallel algorithms for gray-scale digitized picture component labeling on a mesh-connected computer. *Journal of Parallel and Distributed Computing*, 20(1) :56–68, 1994.
- [MM99] Fernand Meyer and Petros Maragos. Morphological scale-space representation with levelings. In *International Conference on Scale-Space Theories in Computer Vision*, pages 187–198. Springer, 1999.
- [NMI79] Makoto Nagao, Takashi Matsuyama, and Yoshio Ikeda. Region extraction and shape analysis in aerial photographs. *Computer Graphics and Image Processing*, 10(3) :195–223, 1979.
- [Soi08] Pierre Soille. Constrained connectivity for hierarchical image partitioning and simplification. *IEEE transactions on pattern analysis and machine intelligence*, 30(7) :1132–1145, 2008.
- [SS93] Jean C Serra and Philippe Salembier. Connected operators and pyramids. In *Image algebra and morphological image processing IV*, volume 2030, pages 65–77. International Society for Optics and Photonics, 1993.
- [SS95] Philippe Salembier and Jean Serra. Flat zones filtering, connected operators, and filters by reconstruction. *IEEE Transactions on image processing*, 4(8) :1153–1160, 1995.