

Projet : Compression basée super-pixels Codage et compression multimédia [HAI809I] Analyse et traitement des images [HAI804I]

Université de Montpellier - FDS
1^{ère} année Master IMAGINE
Jean-Baptiste BES - Thomas CARO - Valentin NOYÉ

03 mars 2024



1 Introduction

Nous étudierons la compression basée sur la segmentation d'une image en groupes de pixels dénommés "super-pixels". Ces super-pixels sont construits sur la base de critères partagés, non seulement de par leur voisinage mutuel, mais par leur similarité en termes de couleur, de structure ou de position. Nous rassemblerons dans ce compte-rendu les concepts de base de cette segmentation en énumérant diverses approches proposées, lesquelles nous permettent de constituer l'état de l'art de cette méthode de compression et nous serviront de base pour l'implémentation de la segmentation en superpixels.

2 Méthodes de segmentation de l'image en superpixels

2.1 Méthodes sélectionnées

Plusieurs méthodes de segmentation ont été sélectionnées nous permettant de comparer efficacement leurs résultats, ordonnées de façon à donner la priorité aux méthodes les plus simples à comprendre et les populaires, soit celles que nous souhaitons pouvoir implémenter. Au minimum, nous espérons pouvoir travailler sur trois de celles-ci. Nous notons K le nombre de décompositions souhaité, et servira notamment d'entrée aux méthodes implémentées dans le but de les comparer.

2.2 SLIC (Simple Linear Iterative Clustering)

L'algorithme SLIC¹ est un des plus populaires dans le domaine et repose sur l'algorithme des K-means appliqué simultanément sur la couleur des pixels et leur position. Les étapes de l'algorithme sont les suivantes :

1. Initialisation des K centres pour le K-means. Ils sont 5-dimensionnels, avec deux composantes spatiales et trois composantes colorimétriques, et sont équidistants d'une certaine distance D .
2. Initialisation de la carte des distances (spatiale et chromatique) aux centres des superpixels avec l'infini et attribution de l'intégralité des pixels au superpixel 0.
3. On calcule pour chaque centre la distance des pixels se trouvant dans un rayon d'une distance 2D du centre. Si elle est plus petite que la précédente, le pixel est réattribué à ce superpixel.
4. Mise à jour des valeurs des centres avec la moyenne de leur pixels attribués.
5. Répétition des étapes 3 et 4 jusqu'à convergence.

Comme on peut le voir ici l'algorithme est très simple à implémenter et en plus assure un bon équilibre entre la compacité des superpixels et leur régularité.

2.3 SLICO (Simple Linear Iterative Clustering Over-Segmentation)

L'ajout d'une sur-segmentation sur l'algorithme SLIC permet de produire de meilleurs résultats en introduisant en plus de la position et de la couleur, un paramètre de compacité permettant de contrôler la sur-segmentation de l'algorithme SLICO,² c'est-à-dire le compromis entre la proximité spatiale et la similarité des couleurs lors de la formation des superpixels.

2.4 Algorithme de Felzenszwalb

L'algorithme de Felzenszwalb,³ aussi bien populaire que SLIC, est un algorithme de segmentation en superpixels basé sur les contours basé sur un graphe pondéré. Son fonctionnement est le suivant :

1. Construction du graphe pondéré, où chaque pixel est un noeud et les arêtes reliant les pixels voisins sont pondérées par leur similarité en termes de couleur.
2. Les régions adjacentes sont fusionnées itérativement si le poids de l'arête qui les relie est inférieur à un seuil prédéfini.
3. Mise à jour des régions après chaque fusion, les propriétés des régions fusionnées (comme la moyenne de couleur, la taille, etc.) sont mises à jour pour refléter les changements.
4. Répétition des étapes 2 et 3 jusqu'à ce que l'on obtienne K régions distinctes.

2.5 ERSS (Entropy-Rate Superpixel Segmentation)

L'algorithme ERSS⁴ se base sur le calcul de l'entropie (de Shannon) des superpixels pour guider la segmentation. Voici son fonctionnement :

1. Initialisation d'une grille régulière, regroupement initial arbitraire ou après un prétraitement tel que SLIC.
2. Calcul de l'entropie des superpixels.
3. Les superpixels ayant une entropie élevée, supérieure à un seuil et donc ayant une grande variabilité de pixels, sont divisés en de plus petits superpixels ayant moins d'entropie.
4. Les superpixels ayant une entropie faible peuvent se fusionner entre voisins sans trop augmenter l'entropie du superpixel généré. La fusion est aussi basée sur la similarité des superpixels, i.e, des superpixels contenant des pixels très différents ne fusionneront pas.
5. On réitère les étapes 2, 3 et 4 jusqu'à obtenir K superpixels.

Cette approche est utile lorsque la texture et la cohérence des régions sont importantes pour l'analyse ultérieure de l'image.

2.6 SEEDS (Superpixels Extracted via Energy-Driven Sampling)

SEEDS⁵ est un algorithme rapide et efficace qui fonctionne en itération afin de décomposer l'image en superpixels tout en minimisant une fonction d'énergie définie sur l'image. Il est populaire pour les applications en temps réel de segmentation, notamment car il permet de choisir des paramètres tels que la taille des super-pixels désirée. Cet algorithme se déroule comme ceci :

1. Division de l'image en grille de blocs où chaque bloc est un superpixel.
2. Calcul de la similarité entre pixels voisins, par couleur par exemple, et prise de décision sur la fusion ou la séparation des superpixels selon un certain seuil défini.
3. Itération visant à minimiser la fonction d'énergie qui dépend de la similarité entre les pixels internes et entre superpixels, ainsi que des paramètres d'entrée.
4. Répétition des précédentes étapes 2 et 3 jusqu'à convergence, c'est à dire jusqu'à la condition reçue en paramètre.

Cet algorithme est donc modulable grâce aux paramètres et peut donc être très rapide selon ce que l'on souhaite faire.

2.7 LSC (Linear Spectral Clustering)

Plus compliqué encore, cet algorithme de partitionnement spectral linéaire⁶ emploie des méthodes de partitionnement spectral tels que N-cuts⁷ (Coupes normalisées) plutôt qu'un simple partitionnement par similarité tel que K-means.

1. Construction de la matrice d'affinité capturant les relations de similarité entre les pixels, basé sur des critères tels que la position et la couleur.
2. Normalisation de la matrice d'affinité pour que ses lignes et ses colonnes aient une somme égale à 1, garantissant que chaque donnée des pixels a une contribution égale à la construction des vecteurs propres.
3. Calcul des valeurs propres et des vecteurs propres de la matrice normalisée, représentent les différentes "fréquences" ou modes de variation dans les données.
4. Les vecteurs propres correspondant aux plus grandes valeurs propres sont sélectionnés pour former une nouvelle représentation des données, avec une dimension réduite.
5. Execution d'un algorithme de partitionnement spectral (N-cuts) en utilisant les vecteurs propres sélectionnés comme caractéristiques, pour regrouper les pixels en superpixels.

2.8 Autres méthodes

Il existe un grand nombre de méthodes, souvent plus compliquées, pouvant être implémentées afin d'arriver à ces fins. En l'occurrence, nous notons les algorithmes Watershed,⁸ ou encore QuickShift⁹ et Turbopixels.¹⁰

3 Compression basée superpixels

Cette segmentation forme la base la plus importante de la compression, quelque soit l'algorithme utilisé, puisqu'elle possède l'avantage de regrouper des pixels d'une même couleur dans de plus gros clusters, ce qui facilite l'application de méthodes classiques de compression telles que le codage prédictif et par plage. Par la suite, elle peut bénéficier d'une réduction de taille aux moyens d'un encodage par dictionnaire tels que ceux de la famille Lempel-Ziv ou encore d'un codage entropique au moyen de Shannon-Fano ou d'Huffman.

Références

- ¹ R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk. Slic superpixels. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34 :2274–2282, Mai 2010.
- ² R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34 :2274–2282, Novembre 2012.
- ³ P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59 :161–181, Septembre 2004.
- ⁴ M.-Y. Liu, O. Tuzel, S. Ramalingam, and R. Chellappa. Entropy rate superpixel segmentation. In *2011 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2097–2104, Juin 2011.
- ⁵ M. Van den Bergh, X. Boix, G. Roig, B. De Capitani, and L. Van Gool. Seeds : Superpixels extracted via energy-driven sampling. *International Journal of Computer Vision*, 111 :298–314, Juillet 2014.
- ⁶ Z. Li and J. Chen. Superpixel segmentation using linear spectral clustering. In *2015 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1356–1363, Juin 2015.
- ⁷ J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22, Août 2000.
- ⁸ Z. Hu, Q. Zou, and Q. Li. Watershed superpixel. In *2015 IEEE International Conference on Image Processing (ICIP)*, pages 349–353, Septembre 2015.
- ⁹ M. Salem, A. Ibrahim, and H. Arafat Ali. Automatic quick-shift method for color image segmentation. In *2013 8th International Conference on Computer Engineering Systems (ICCES)*, pages 245–251, Novembre 2013.

- ¹⁰ A. Levinstein, A. Stere, K. Kutulakos, D. J. Fleet, S. J. Dickinson, and K. Siddiqi. Turbopixels : Fast superpixels using geometric flows. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31 :2290–2297, Mai 2009.