Décomposition d'images en superpixels et transfert de couleurs

Nathan Lichtlé

Sous la supervision de Vinh-Thong Ta et Rémi Giraud

Juin-Juillet 2018

Université de Bordeaux
Laboratoire Bordelais de Recherche en Informatique (LaBRI)
Équipe Pictura

Plan

Environnement de recherche

Université de Bordeaux

LaBRI

SCALP (2018), SPM et SCT (2017)

Comprendre $\rightarrow$ Implémenter $\rightarrow$ Améliorer

Définitions et motivations

Image

Image de hauteur h et de largeur w

Taille = $h \times w$ pixels

Espace RGB (entre 0 et 255)

Espace CIELAB, plus adapté pour traiter des images

Superpixels - Définition

Partition $\{S_1, \ldots, S_k\}$ d'une image I

- Adhérents aux contours
- Compacts, uniformes, connexes
- Cohérents
- Rapides à calculer

Superpixels - Exemple de décomposition

Décomposition d'une image en superpixels pour différentes tailles.

Exemple de transfert de couleur

Image cible

Image source

Résultat du transfert

Exemple d’un transfert de couleur avec SCT.

Motivations

Décomposition en superpixels = base de nombreux algorithmes (segmentation, reconnaissance d'objets, tracking)

Transfert de couleurs (restauration, recolorisation)

Extension 3D

Décomposition en superpixels

Simple Linear Iterative Clustering - SLIC

Adapté d'un algorithme de k-moyennes

Initialisation des centres $c_1^{(1)}, \ldots, c_k^{(1)}$

Répéter jusqu'à convergence :

1. Assigner chaque pixel au centre le plus proche

$$S_i^{(t)} = \{p \ : \ ||p - c_i^{(t)}|| \leq ||p - c_j^{(t)}|| \ \forall j = 1, \ldots, k\}$$

2. Mettre à jour les centres

$$c_i^{(t+1)} = \frac{1}{|S_i^{(t)}|} \sum_{p \in S_i^{(t)}} p$$

Rendre les superpixels connexes si nécessaire

Simple Linear Iterative Clustering - SLIC

Pixel $p = (l, a, b, x, y)$

$$d_c(p_1, p_2) = \sqrt{(l_1 - l_2)^2 + (a_1 - a_2)^2 + (b_1 - b_2)^2}$$

$$d_s(p_1, p_2) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

$$D(p_1, p_2) = \sqrt{\frac{d_s(p_1, p_2)}{S} + \frac{d_c(p_1, p_2)}{m}}$$

Simple Non-Iterative Clustering - SNIC

Version non-itérative de SLIC

File de priorité

Mise à jour en direct

Superpixels connexes

Linear Spectral Clustering - LSC

Pareil que SLIC mais en 10 dimensions

$p = (l_1, l_2, a_1, a_2, b_1, b_2, x_1, x_2, y_1, y_2)$

Meilleurs résultats mais plus lent (20 itérations)

Amélioration de SLIC et LSC

10 dimensions : $F_k = [l_k^1, l_k^2, a_k^1, a_k^2, b_k^1, b_k^2]$

Voisinage : $\mathcal{P}(p)$ = carré de côté $2n+1$ centré sur p

Chemin linéaire $\mathbb{P}_p^k$ entre p et C_k

$$D_c(p, C_k) = \sum_{q \in \mathcal{P}(p)} (F_q - F_{C_k})^2 w_{p,q}$$

$$d_c(p, C_k) = \lambda D_c(p, C_k) + (1-\lambda)\frac{1}{\mathbb{P}_p^k} \sum_{q \in \mathbb{P}_p^k} D_c(q, C_k)$$

Amélioration de SLIC et LSC

Carte de contours $\mathcal{C}$

Distance spatiale d_s en 10 dimensions

$$d_{\mathcal{C}}(p, C_k) = 1 + \gamma \max_{q \in \mathbb{P}_p^k} \mathcal{C}$$

Distance finale à minimiser avec SLIC :

$$D(p, C_k) = \left(\frac{d_c(p, C_k)}{m^2} + \frac{d_s(p, C_k)}{S^2} \right) d_{\mathcal{C}}(p, C_k)$$

Comparaison des différents algorithmes


SLIC


SNIC


LSC


SCALP

Exemples de décompositions pour $K = 300$ superpixels

Un nouvel algorithme : SNALP

Fusion de SNIC et SCALP

SNIC > SLIC et SCALP > SLIC donc SNALP > SCALP ?

SCALP = SLIC + distance SCALP

SNIC = SLIC non-itératif

SNALP = SCALP non-itératif, ou SNIC + distance SCALP

Métriques

Global Regularity measure (GR)

régularité + uniformité des superpixels

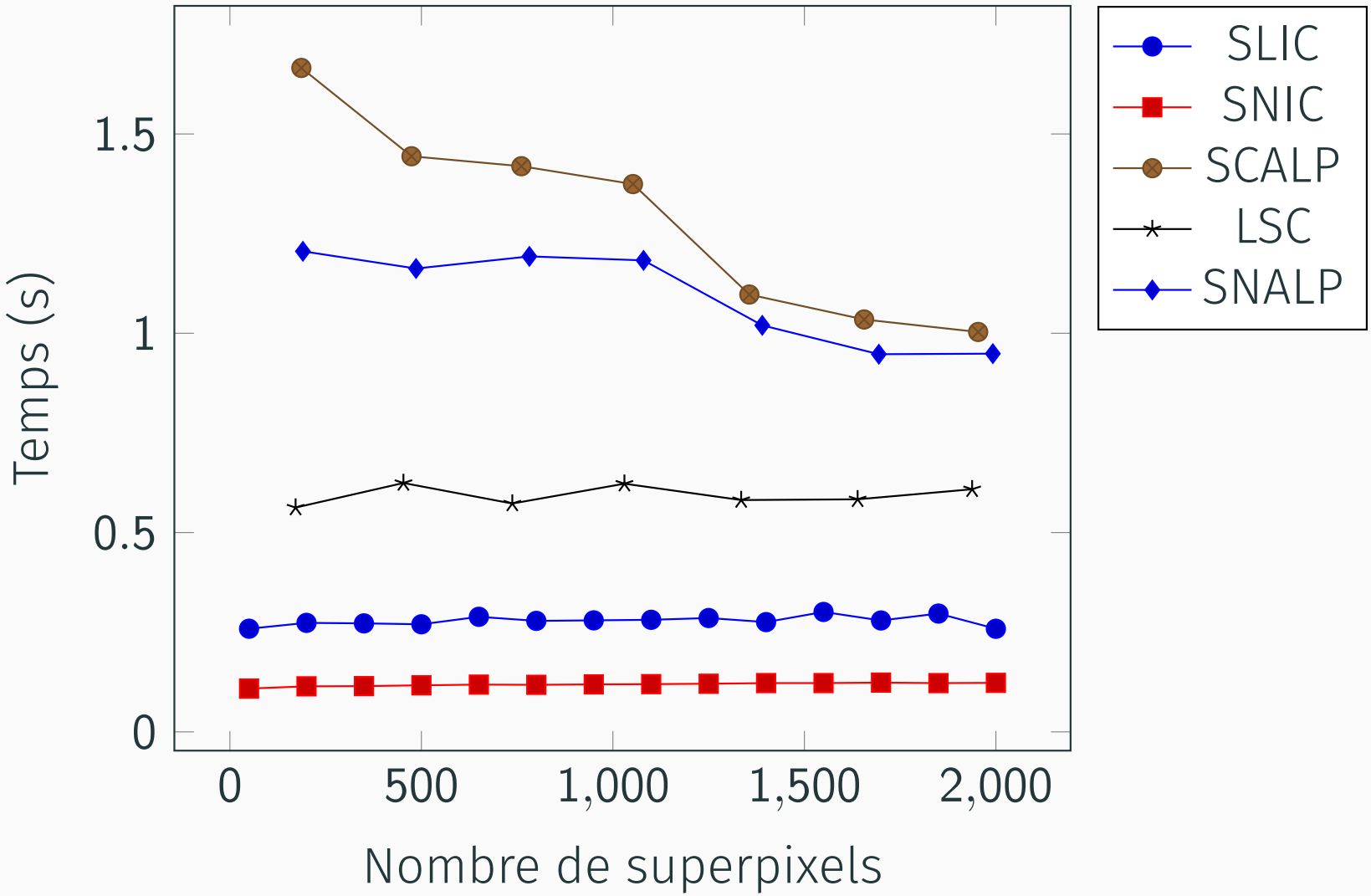
Achievable Segmentation Accuracy measure (ASA)

adhérence aux contours (terrains de vérité)

Berkeley Segmentation Dataset (BSD)

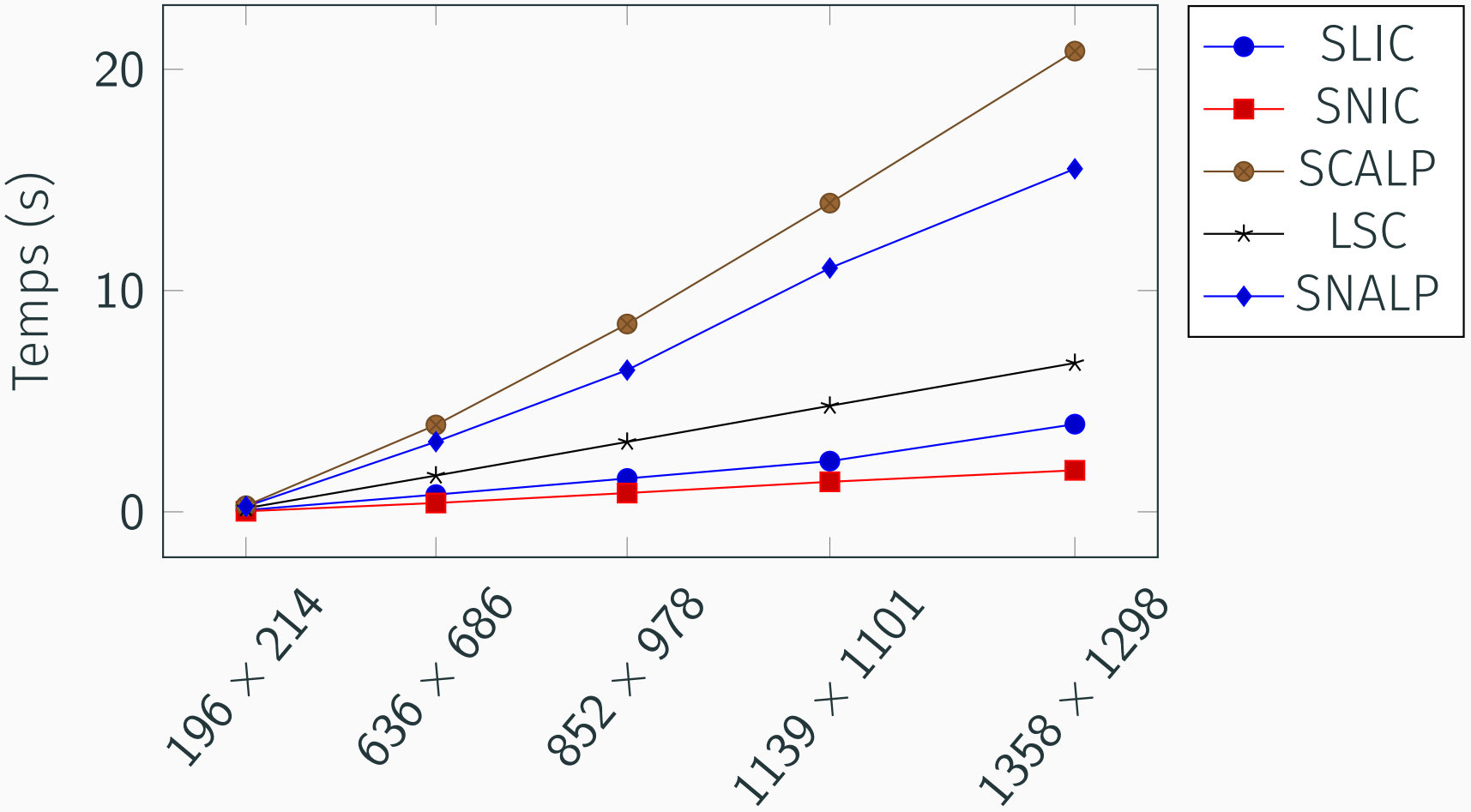
200 images + au moins 5 terrains de vérité par image

Résultats - Temps d'exécution et linéarité



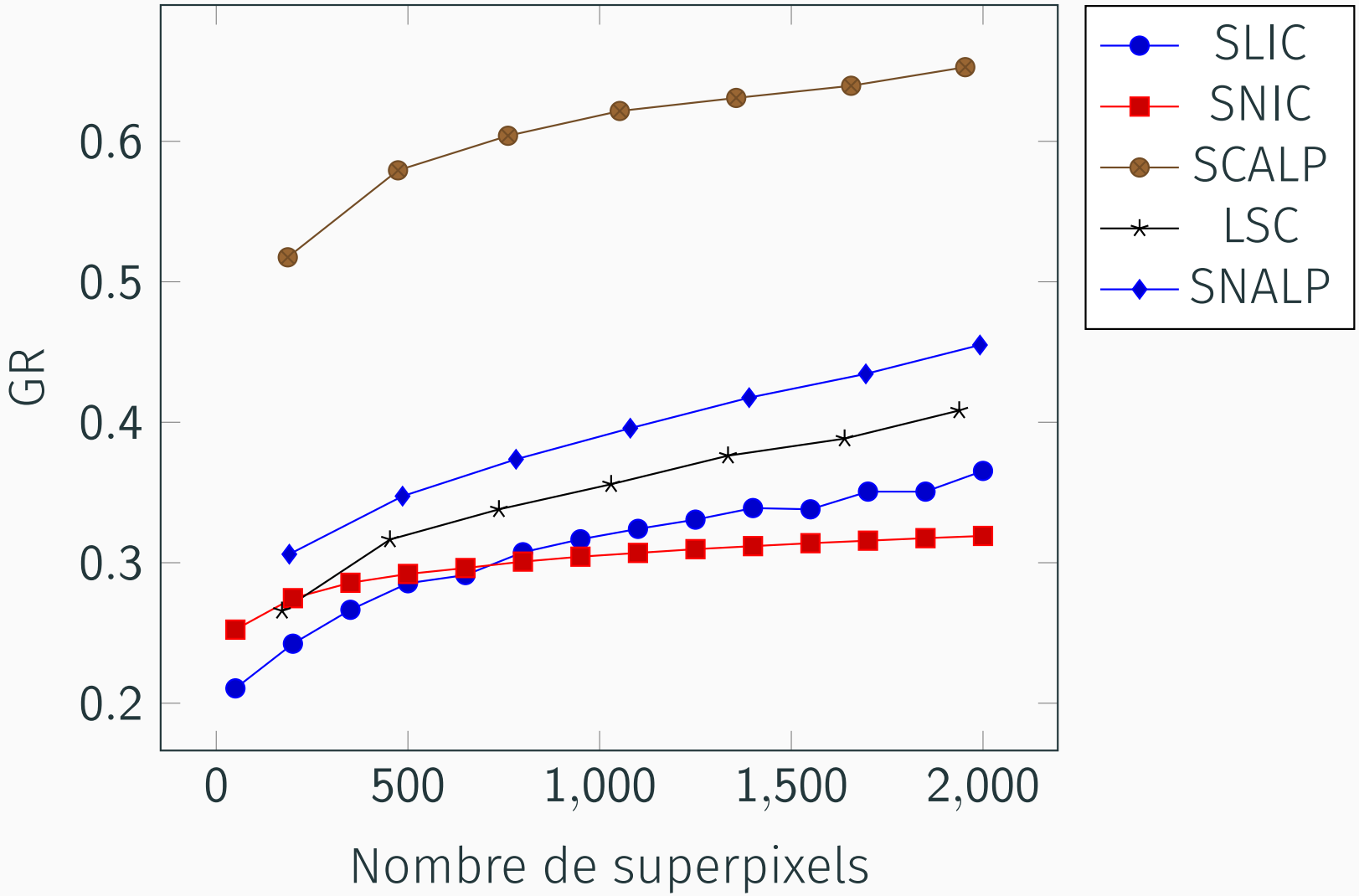
Temps d'une segmentation pour différents nombres de superpixels

Résultats - Linéarité



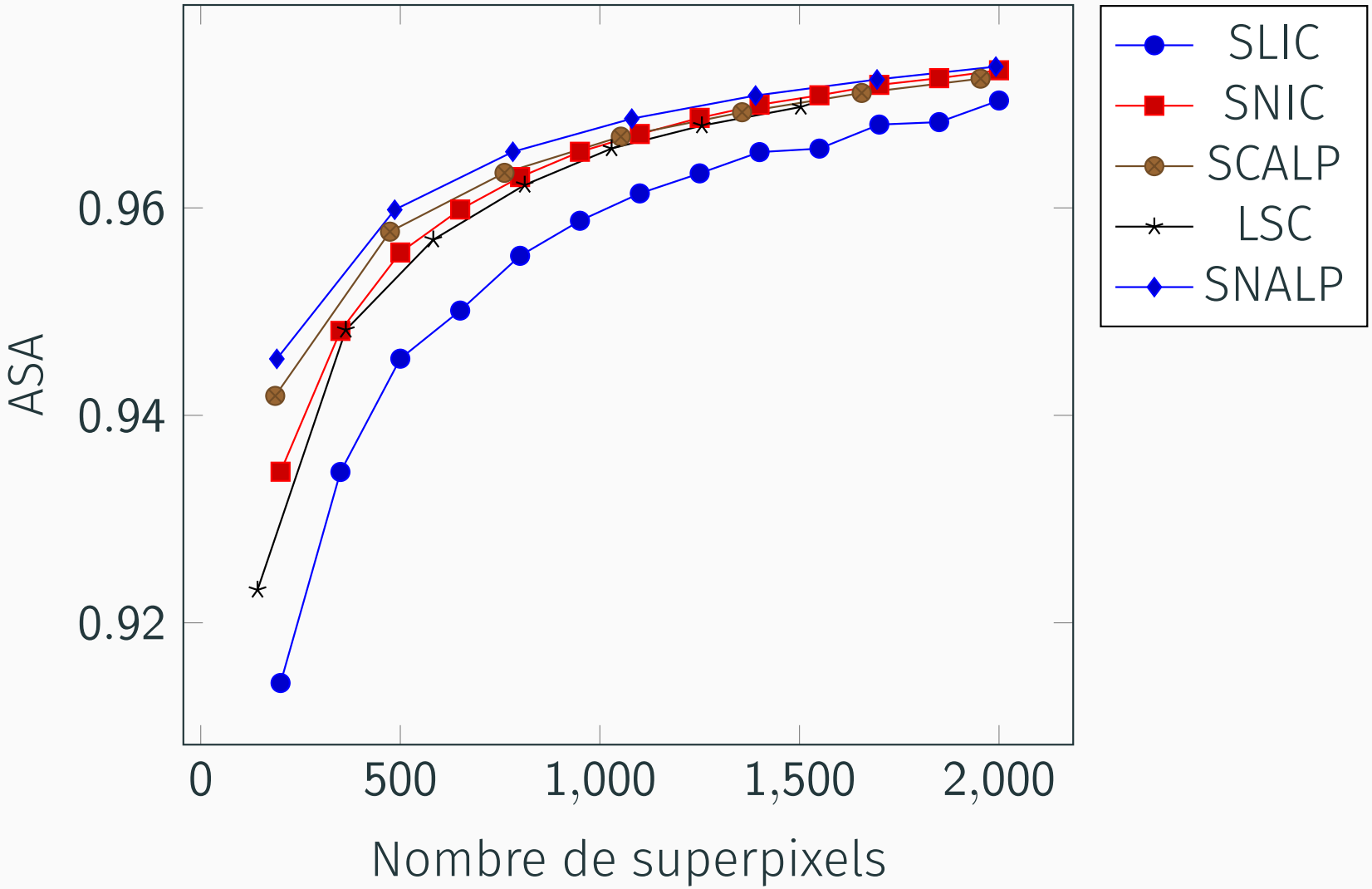
Temps d'une segmentation pour différentes tailles d'images, pour $K = 400$ superpixels

Résultats - GR



Mesure GR de segmentations pour différents nombres de superpixels

Résultats - ASA



Mesure ASA de segmentations pour différents nombres de superpixels

Transfert de couleurs

Algorithme basique

1. RGB $\rightarrow l\alpha\beta$
2. Centrer-réduire puis faire l'opération inverse

$$\forall (l, \alpha, \beta) \in B, \quad l \leftarrow \frac{l - \mu_B^l}{\sigma_B^l} \sigma_A^l + \mu_A^l$$

$$\alpha \leftarrow \frac{\alpha - \mu_B^\alpha}{\sigma_B^\alpha} \sigma_A^\alpha + \mu_A^\alpha$$

$$\beta \leftarrow \frac{\beta - \mu_B^\beta}{\sigma_B^\beta} \sigma_A^\beta + \mu_A^\beta$$

3. $l\alpha\beta \rightarrow$ RGB

Exemple






Image cible

Image source

Résultat de la fusion

Exemples de transferts de couleurs avec l'algorithme basique

Carte ANN (approximate nearest neighbor)

Image A, décomposition $\mathcal{S} = \{S_1, \ldots, S_p\}$

Image B, décomposition $\mathcal{R} = \{R_1, \ldots, R_q\}$

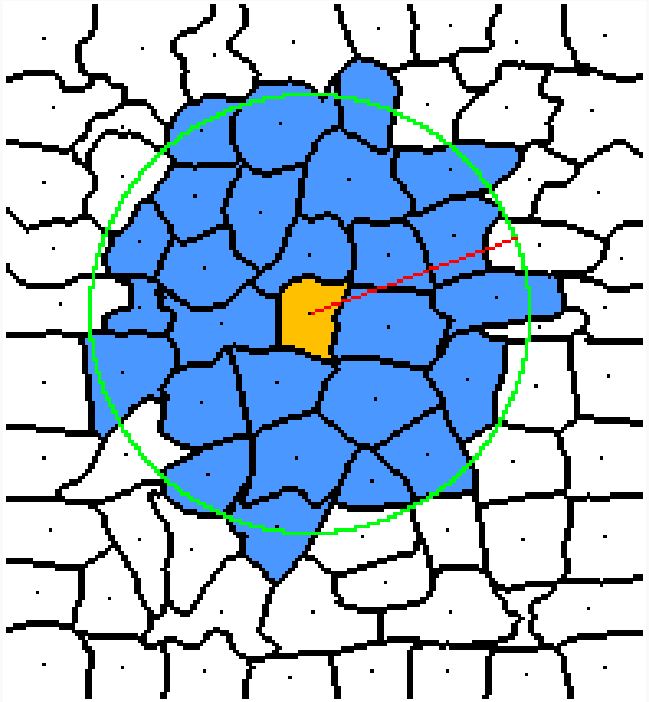
$$\text{Distance } D(S_i, R_j) = \sqrt{\sum_{\text{col} \in \{Red, Green, Blue\}} [E_{\text{col}}(S_i) - E_{\text{col}}(R_j)]^2}$$

Tableau de correspondance M (plus proches voisins)

$$\text{Minimiser } \sum_{i=1}^{p} D(S_i, M(S_i))$$

Superpatch

$$\mathbf{S_i} = \{S_k \mid ||c_i - c_k|| \leq R\}$$



Exemple d'un superpatch $\mathbf{S_i}$ (en bleu) centré autour d'un superpixel S_i (en jaune), avec le rayon R représenté par une ligne rouge.

SuperPatchMatch - SPM

But : trouver une carte ANN de A vers B

$$D(A_i, B_j) = \frac{\sum_{A_{i'} \in \mathbf{A}_i} \sum_{B_{j'} \in \mathbf{B}_j} w(A_{i'}, B_{j'}) d(A_{i'}, B_{j'})}{\sum_{A_{i'} \in \mathbf{A}_i} \sum_{B_{j'} \in \mathbf{B}_j} w(A_{i'}, B_{j'})}$$

Initialisation : carte ANN aléatoire

Itérer :

1. Propagation
2. Recherche aléatoire

SuperPatchMatch - SPM

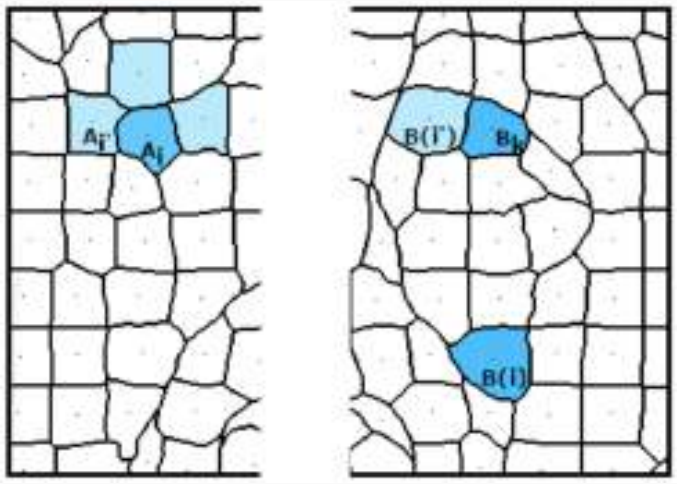
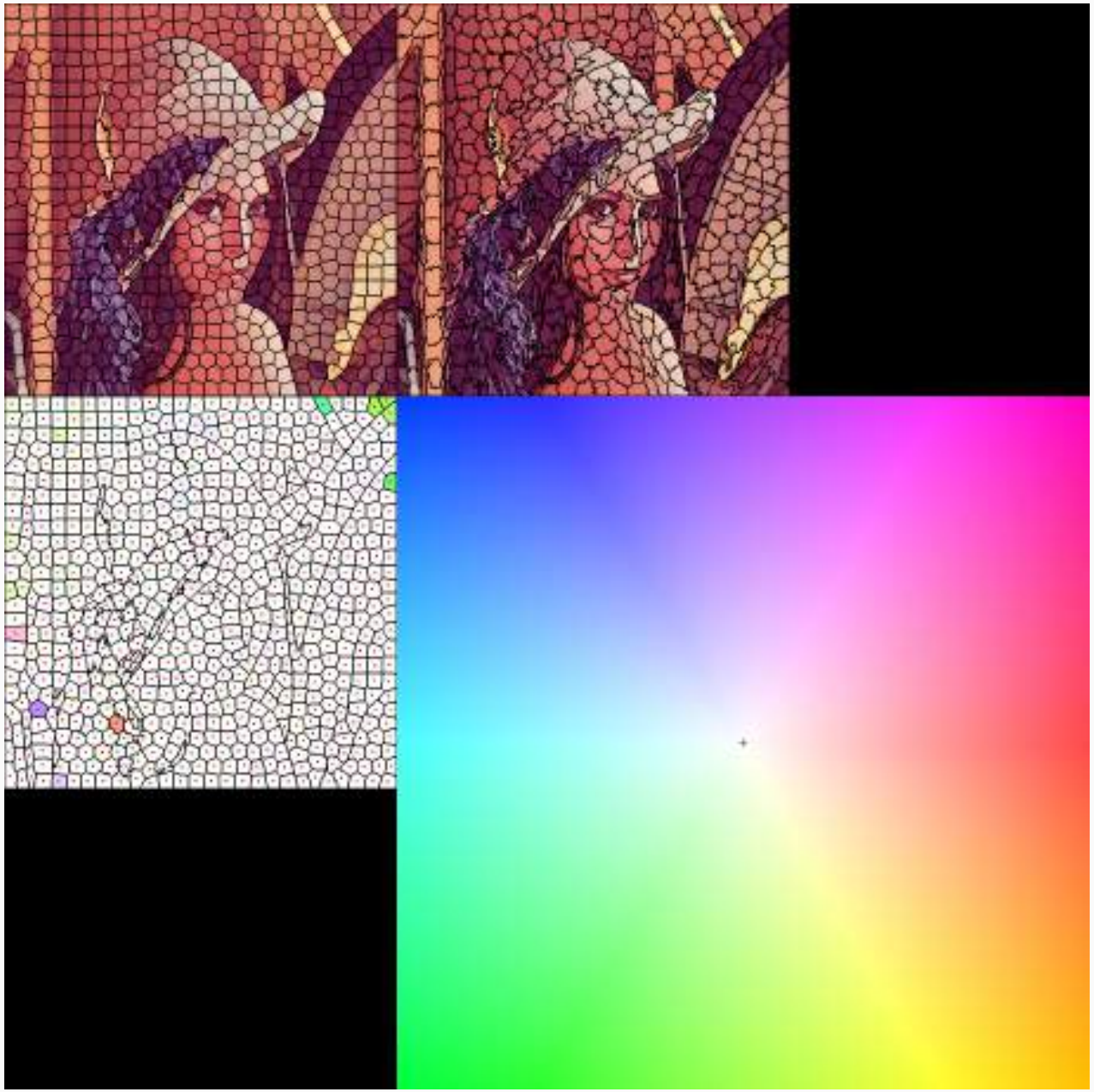


Illustration de l'étape de propagation. A_i est pour l'instant associé à $B(i)$. Ses voisins déjà rencontrés (dans le cas d'une itération paire) sont considérés pour fournir des nouveaux candidats. Un voisin $A_{i'}$ est associé à $B(i')$ ce qui mène au candidat B_k, le voisin de $B(i')$ avec la position relative la plus similaire à celle entre A_i et $A_{i'}$.

Résultats



Superpixels Color Transfer - SCT

Décomposition $\rightarrow$ carte ANN $\rightarrow$ transfert

$$A(p) = \frac{\sum_{A_j} w(p, A_j) C_{B(j)}}{\sum_{A_j} w(p, A_j)}$$

Résultats


Image source


Image cible


Carte ANN


Résultat de SCT

Transfert de couleur utilisant SCT, avec l'ANN calculé par SPM.

Conclusion

Implémentations & SNALP

La suite :

Améliorer les performances

Généralisation aux supervoxels

Prendre en compte les textures