

Poster

- introduction / context / motivatio
- methode
- resultats
- conclusion
- (super pixels magnier thibault)

Super-pixels

But

Empacter des groupe de pixels similaires entre eux pour former un unique pixels, cela peut servir pour la segmentation et pour la compression également

super-pixels -> resume les information d'une images

reduit le nombre d'entités à traiter

SLIC / SNIC

de plus en plus populaires pour la vision par ordinateur, cependant il n'y a pas beaucoup d'algo qui donne super pixels compacts avec efficacités (temps linéaire).

SLIC rassemble les pixels en s'aidant combinant les couleurs et le plan de l'image dans un espace 5 dimension

paramètre -> nombre de super_pixels

slic meilleur que 4 autres algo avant lui

Super pixels

- Enlève les redondances de l'image
- Permettent de reduire grandement les éléments à traiter
- Pratique pour l'estimation de profondeur dans une image
- Pratique pour la segmentation d'image
- Pratique pour la localisation d'objets

Remarque : pour être utiles ils doivent fournir un segmentation de qualité et être rapide à l'exécution (surtout si on veut les utiliser pour de la vision par ordinateur)

Algorithme	Graphe	Graphe	Graphe	Gradient	Gradient	Gradient	Gradient	Gradient
Propriété	GS04	NC05	SL08	WS91	MS02	TP09	QS09	SLIC
Ctrl nb de super pixels	Non	Oui	Oui	Non	Non	Oui	Non	Oui
Ctrl Compacité	Non	Oui	Oui	Non	Non	Oui	Non	Oui
Complexité $O(\cdot)$	$N \cdot \log. N$	$N^{\frac{3}{2}}$	$N^2 \cdot \log. N$	$N \cdot \log. N$	N^2	N	dN^2	N
Paramètres	2	1	3	1	3	1	2	1

Méthode par graphe

- Pixels \rightarrow nœud du graphe
- Arêtes \rightarrow pondérées proportionnellement à la similarité entre les pixels
- Segments \rightarrow extraits en minimisant une fonction de coût sur le graphe

Méthode par gradient

- On part ensemble de super-pixels uniformes
- On améliore les super-pixels de manière itératives

SLIC : Simple Linear Iterative Clustering

- Significativement plus efficace en segmentation et en temps d'exécution que les autres algorithmes
- Rassemblement des pixels en 5D (l, a, b, x, y)
- Garantie la similarité de couleur et la proximité des pixels
- (l, a, b) Couleur CIELAB \rightarrow Plus conforme à la perception des écarts de couleur par les humain (même pour des petites distances)
- (x, y) Position \rightarrow Normalisé pour ne pas donner trop d'importance aux grande distances

Mesure de distances

- Pour K superpixels sur une image de N pixels on à $\frac{N}{K}$ pixels par superpixels
- On place un centre sur chaque intervalle de grille $S = \sqrt{\frac{N}{K}}$
- L'espace de recherche pour les pixels d'un super pixels est dans la zone $2S \times 2S$ autour du centre
- $d_{lab} = \sqrt{(l_k - l_i)^2 + (a_k - a_i)^2 + (b_k - b_i)^2}$
- $d_{xy} = \sqrt{(x_k - x_i)^2 + (y_k - y_i)^2}$
- $Ds = d_{lab} + \frac{m}{S} d_{xy}$
- On peut contrôler la compacité des superpixels avec la variable $m \in [1, 20]$
- Plus m est grand plus les supers pixels seront compacts

Algorithmes

1. Initialize cluster centers $C_k = [l_k, a_k, b_k, x_k, y_k]^T$ by sampling pixels at regular grid steps S .
2. Perturb cluster centers in an $n \times n$ neighborhood, to the lowest gradient position.
3. **repeat**
4. **foreach** cluster center C_k **do**

5. Assign the best matching pixels from a $2S \times 2S$ square neighborhood around the cluster center according to the distance measure.
6. **end for**
7. Compute new cluster centers and residual error E
{ $L1$ distance between previous centers and recomputed centers}
8. **until** $E \leq \text{threshold}$
9. Enforce connectivity.