

# INF555 TD6

## Synthèse de texture

— avec une méthode rapide pour les plus proches voisins —

Frank NIELSEN  
nielsen@lix.polytechnique.fr

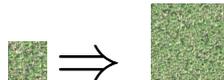
19 Octobre 2011

### 1 L'algorithme d'Efros-Leung

Implanter l'algorithme d'Efros-Leung pour la synthèse de texture, en utilisant le voisinage en forme de  $L$ . On implantera d'abord la méthode naïve pour calculer le plus proche voisin en temps linéaire. Faire des tests de qualité/temps de calcul pour un voisinage de côté  $2s + 1 = 3$  puis 5 et enfin 11. On pourra utiliser l'image  $64 \times 64$  couleur `grass.ppm` comme source, et créer une texture de taille  $128 \times 128$ . Vous pouvez également choisir vos propres images de texture en regardant par exemple sur le site :

<http://www.ux.uis.no/~tranden/brodatz.html>

Bonus: montrer la progression de la synthèse de texture sous Processing



### 2 Accélération du calcul du plus proche voisin

Afin de rendre *pratique* cet algorithme, nous nous proposons d'accélérer les requêtes de plus proches voisins. Pour cela nous allons implanter un arbre de recherche *metric ball tree* pour un ensemble de  $n = |\mathcal{S}|$  points  $\mathcal{S}$  comme suit:

1. Tirer un point aléatoire  $p_1 \in \mathcal{S}$ , puis calculer la distribution des distances de  $p_1$  aux autres points de  $\mathcal{S}$ . Choisir aléatoirement ensuite un point  $p_2 \neq p_1 \in \mathcal{S}$  uniformément en fonction de la distribution des distances. (C'est l'initialisation d'un 2-means++).

2. Découper  $\mathcal{S}$  en deux paquets  $\mathcal{S}_1$  (les points plus proches de  $p_1$ ) et  $\mathcal{S}_2$  (les points plus proches de  $p_2$ ). On note  $r_1 = \max_{s \in \mathcal{S}_1} \|p_1 - s\|$  et  $r_2 = \max_{s \in \mathcal{S}_2} \|p_2 - s\|$ , les rayons des boules correspondantes.

3. Si  $|\mathcal{S}_1|$  a plus de  $n_0 = 25$  points créer un fils gauche en appelant récursivement l'étape 1. De même, si  $|\mathcal{S}_2| \geq n_0$ .

Notez que les points sont stockés aux feuilles uniquement. Les noeuds internes ont les deux boules centrées aux centroides uniquement. Lors d'une requête  $\text{NN}(q)$  pour un point  $q$ , on initialise  $r_q \sim +\infty$  et on parcourt l'arbre de la racine aux feuilles. Quand on arrive à une feuille, on calcule le plus proche voisin par la méthode linéaire naïve et on met à jour  $\text{NN}(q)$  et  $r_q = \|q - \text{NN}(q)\|$  éventuellement si on trouve un meilleur plus proche voisin. Pour éviter de visiter tous les fils d'un noeud, on calcule la distance minimale de  $q$  aux boules. Si cette distance est plus grande que le rayon  $r_q$  actuel, on évite ainsi de visiter les noeuds du sous-arbre correspondant (*pruning*).

Observer le gain de temps pour la synthèse de texture en utilisant cette structure de données.