

SYNTHESE ET ADVECTION DE TEXTURES

1 Problématique :

La synthèse de texture consiste à générer une large texture à partir d'un petit échantillon (exemplaire). L'intérêt est de créer des textures sans limitation de taille et de motif.

Ces textures peuvent être utilisées ensuite pour être appliquées à des modèles en 3D, sur différentes surfaces mais encore, une fois animées, à modéliser des mouvements fluides.

2 Algorithme :

De nombreux algorithmes ont été élaborés à ce jour. Dans ce projet, nous nous basons sur les travaux de Sylvain Lefebvre, ancien ENSIMAG.

Les modèles de textures doivent être raccordable sur les bords (images tores).

Tout d'abord, il faut savoir que pour chaque image sur laquelle on travaille, on utilise en réalité les matrices des coordonnées des pixels situés dans le modèle exemplaire E de cette image.

Ainsi, chaque case de cette matrice $S(x,y)$ contient un pointeur $P \in \mathbb{R}^2$ tel que la couleur au pixel (x,y) de l'image est celle du pixel de l'exemplaire pointé par $P=S(x,y)$ c'est à dire $E[P]=E[S(x,y)]$.

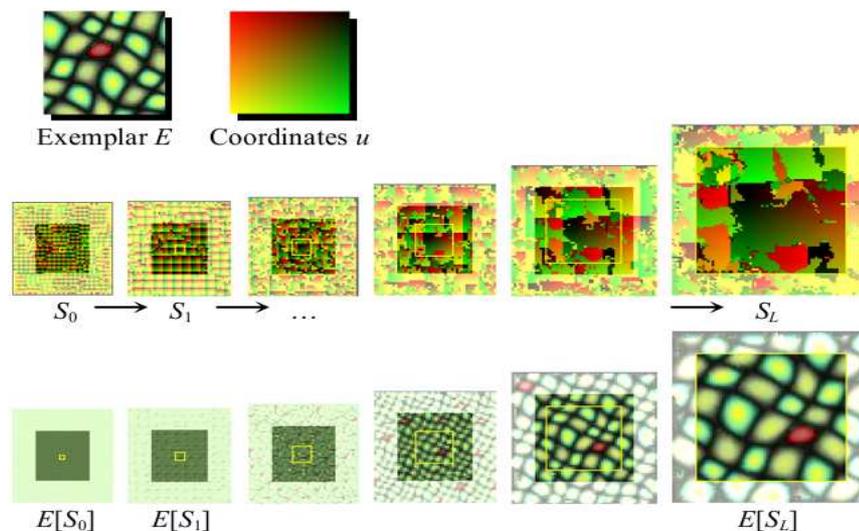


figure 1 : Pyramide d'images de S et l'image correspondant

Notre algorithme de synthèse utilise une pyramide d'images. Cela permet d'effectuer au départ des calculs sur des petites images (donc plus rapides). Les bas niveaux de la pyramides vont donner la structure globale de l'image finale, tandis que les hauts niveaux affinent le résultat.

Pour chaque étage de la pyramide, notre algorithme de synthèse se décompose en 3 étapes :

Une étape d'Upsampling : On agrandit l'image d'un facteur 2 pour chaque dimension. C'est ce qui fait passer au niveau suivant dans la pyramide.

Une étape de Jittering : On mélange localement les pixels de l'image permettant de donner une structure qui ne soit pas des copies conformes de l'exemplaire.

Une étape de Correction : Cette étape est de loin celle qui prend le plus de temps, c'est elle qui va permettre de générer une image qui ressemble à l'exemplaire de départ.

On parcourt l'image pixel par pixel (dans le sens de la lecture) et pour chaque pixel nous appliquons la correction. Cette correction considère le voisinage de ce pixel et recherche dans l'image

exemplaire le pixel qui a le voisinage le plus proche. Elle remplace alors le pixel de l'image à corriger par le pixel de l'exemplaire de voisinage le plus proche. A chaque correction, on tient compte des corrections des pixels précédents, ce qui permet une vitesse de convergence plus importante.

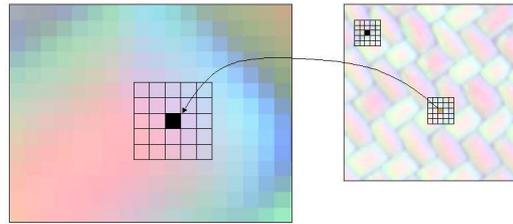


figure 2 : Voisinage kxk

Le voisinage $k * k$ (k impair) du pixel (x,y) est l'image composée des $k * k$ pixels dont le pixel (x,y) est le centre.

Pour les bords, par exemple, si le pixel est situé sur le bord gauche de l'image, on considère que son voisin de gauche est situé sur le bord droit de l'image (et c'est pour cette raison qu'elle doit être raccordable aux bords pour avoir une certaine cohérence).

Pour la recherche des voisinages, on a utilisé la librairie ANN qui permet d'accélérer très considérablement les temps de calcul.

3 Advection :

Le but de l'advection est de donner à la texture une forme suivant un champ de vecteurs (un angle et une norme pour chaque pixel).

Cela s'effectue au cours de la correction lors de la synthèse .

Le principe est le suivant :

Au lieu de considérer un voisinage carré droit, on considère un voisinage tourné et agrandi selon l'angle et la norme associé à ce pixel.

4 Animation :

Pour animer la texture, on utilise un tableau de champs de vecteur (un champ de vecteur par unité de temps). Cette fois-ci, un champ de vecteur (angle et norme pour chaque pixel) donne l'allure du mouvement.

En pratique, on commence par synthétiser la texture. On déplace ensuite chaque pixel de cette image selon le vecteur lui correspondant puis on effectue une passe de correction. On recommence ainsi de suite pour chaque unité de temps.

La passe de correction permet de garder la cohérence de la texture par rapport à l'exemplaire.

5 Analyse :

Même si nos résultats donnent des images de qualité semblable à celles de Sylvain Lefebvre, notre algorithme, n'ayant pas été implémenté en calcul parallèle, est long à s'exécuter tandis que celui de Sylvain Lefebvre s'effectue en temps réel.

Il est intéressant de noter également que la qualité de la texture obtenue dépendant très fortement des réglages du synthétiseur et du motif de départ.

Le paramètre le plus important est la taille du voisinage. C'est elle qui donne au résultat la ressemblance avec le modèle. Il faut que le k soit suffisamment important pour pouvoir capturer un motif de base dans la texture. L'intérêt de travailler avec une pyramide d'images tout en gardant la même taille de voisinage et que sur les petites images, le voisinage capturera des plus gros motifs car il sera plus gros en proportion. Ainsi on capture mieux la texture dans les bas niveaux de la pyramide. Attention toutefois, le temps de calcul augmente avec k .