

Cet énoncé s’inspire largement de l’article “Image Quilting for Texture Synthesis and Transfer” par Alexei A. Efros et William T. Freeman paru dans les actes de la conférence SIGGRAPH 2001, et nous est parvenu via Basile Sauvage et Guillaume Gillet. Le travail sera à réaliser, de préférence, par groupe de trois ou quatre.

1 Le contexte: la synthèse de textures

On considère le problème suivant, dans le domaine de la synthèse d’images: remplir une surface quelconque avec une texture naturelle. Une texture est définie à l’aide d’un échantillon, généralement une image carrée d’une taille réduite. L’idée de base est très simple: on veut paver la surface à remplir avec l’échantillon.

La figure 1 illustre 3 façons différentes de mettre en œuvre l’idée de base. Dans tous les cas, on utilise des “blocs” qui sont des sous-images de l’échantillon; par exemple, l’échantillon sera découpé en 4 ou en 9 blocs. Dans l’exemple de la figure, la surface à remplir est un carré ayant 3 fois les dimensions de l’échantillon.

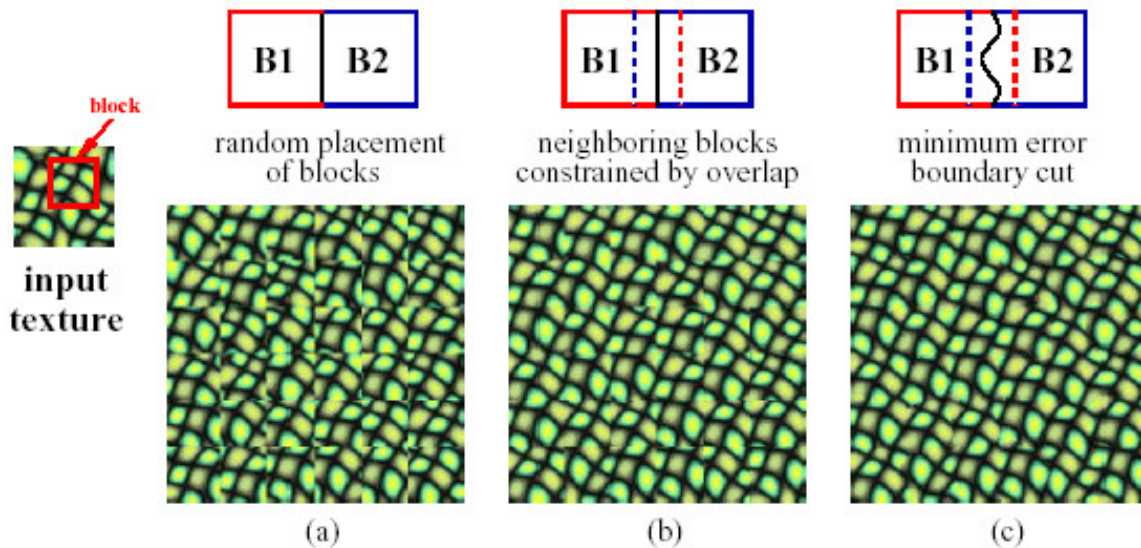


Figure 1: Illustration de 3 algorithmes de raccordement: juxtaposition de blocs choisis de façon aléatoire; blocs recouvrants choisis avec un critère d’erreur; recouvrement avec coupe optimale.

Dans l’**algorithme (a)** les blocs sont définis sans recouvrement. Ils sont choisis aléatoirement. Dans l’**algorithme (b)** les blocs sont définis avec un recouvrement. On remplit en

choisissant un par un des blocs minimisant un critère d'erreur par rapport aux blocs déjà placés. On utilise une coupe droite.

Dans l'**algorithme (c)** les blocs sont définis avec un recouvrement. On cherche une "coupe optimale", qui est un chemin dans la zone de recouvrement des 2 blocs assurant le meilleur raccordement. On choisit ensuite le bloc de coupe optimale d'erreur minimale.

2 Problème posé

Dans la zone de recouvrement de deux images, trouver une coupe optimale. Il s'agit d'un problème d'optimisation combinatoire qui peut se résoudre par des algorithmes très différents. Nous allons l'aborder par la méthode de programmation dynamique. D'une façon un peu surprenante, il peut aussi être modélisé comme la recherche d'un plus court chemin dans un graphe.

3 Critère d'erreur

Soient B^1 et B^2 les bandes de recouvrement de deux blocs voisins. Nous utilisons comme critère d'erreur entre 2 blocs la différence cumulée entre pixels :

$$E = \sum_{i,j} \|B_{i,j}^2 - B_{i,j}^1\| = \sum_{i,j} e_{i,j}.$$

L'erreur $e_{i,j}$ entre deux pixels est la norme 2 des écarts sur les canaux RGB. Pour l'algorithme (b) les indices parcourent toute la bande de recouvrement. Pour l'algorithme (c) les indices parcourent toute la coupe optimale.

4 Coupe optimale

L'article qui a motivé ce sujet contient les précisions suivantes:

To find the minimal vertical cut through this surface we traverse $e(i = 2..N)$ and compute the cumulative minimum error E for all paths:

$$E_{ij} = e_{ij} + \min(E_{i-1,j-1}, E_{i-1,j}, E_{i-1,j+1})$$

5 Travail demandé

1. Expérimentez les algorithmes (a) et (b) fournis avec différentes images, et différents paramètres.
2. Implémentez une variante de l'algorithme (a). Un générateur aléatoire de permutation de blocs permet d'équilibrer le nombre d'occurrences de chaque bloc.
3. Que représentent e et E ?

4. Quel est le rapport entre le coût de la coupe optimale et les E_{ij} ?
5. Clarifiez la notion de coupe optimale sur l'exemple suivant :

$$e = \begin{array}{|c|c|c|c|} \hline 1 & 3 & 2 & 1 \\ \hline 2 & 1 & 2 & 3 \\ \hline 1 & 3 & 4 & 2 \\ \hline 2 & 4 & 3 & 1 \\ \hline 4 & 3 & 1 & 2 \\ \hline \end{array}$$

6. Estimez la complexité d'une solution récursive naïve pour le calcul de la coupe optimale.
7. Implémentez une solution récursive sans calcul redondant.
8. Implémentez une solution itérative.
9. Question complémentaire : quelle est la complexité globale de l'algorithme (c) en prenant en compte tous les paramètres ?

6 Dossier à rendre

Envoyez à votre enseignant par courriel une archive contenant un répertoire aux noms de votre groupe (3 ou 4 personnes de préférence) dans lequel se trouve :

- les codes source **propres et commentés** du `permuteur`, du `raccordeur_recuratif` et du `raccordeur_iteratif`.
- Un compte-rendu de **2 pages maximum au format pdf** (les figures et tableaux ne comptant pas dans ces deux pages) contenant :
 - une description succincte de vos programmes.
 - d'amples comparaisons de la qualité des résultats obtenus pour les différentes solutions : qualité de texture, vitesse d'exécution.