

INFO-2050 : Mise en page automatique d'une bande dessinée

Rapport

Maxime GOFFART
180521

Olivier JORIS
182113

2019 - 2020

3.1 Approche exhaustive pour déterminer le sillon optimal

Soient n , la hauteur de l'image, et m , la largeur de celle-ci. Une approche exhaustive pour déterminer le sillon optimal serait d'envisager tous les sillons possibles arrivant à un pixel de la dernière ligne de l'image et de prendre celui avec le coût minimal en énergie. Pour un pixel fixé du sillon, on a 3 possibilités pour le prochain : celui directement en-dessous, celui en-dessous à gauche ou celui en-dessous à droite. Etant donné que la dernière ligne, comme toutes les autres, comporte m pixels, on a m potentiels pixels faisant partie du sillon optimal pour celle-ci. La complexité d'une telle approche est donc en $\Theta(m * 3^n)$ et est donc exponentielle par rapport à n .

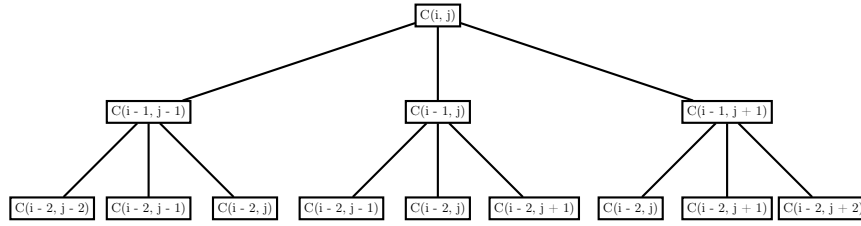
3.2 Formulation récursive du coût d'énergie du sillon optimal

Soit $C(i, j)$, le coût en énergie du sillon optimal s'arrêtant au pixel (i, j) avec $1 \leq i \leq n$ et $1 \leq j \leq m$.

$$\text{On a : } C(i, j) = \begin{cases} E(i, j) & \text{si } i = 1 \\ E(i, j) + \min\{C(i-1, j-1), C(i-1, j)\} & \text{si } j = m \text{ et } i > 1 \\ E(i, j) + \min\{C(i-1, j), C(i-1, j+1)\} & \text{si } j = 1 \text{ et } i > 1 \\ E(i, j) + \min\{C(i-1, j-1), C(i-1, j), C(i-1, j+1)\} & \text{sinon.} \end{cases}$$

3.3 Graphe des appels récursifs

Le graphe ci-dessous représente les premiers appels récursifs pour trouver la fonction de coût du sillon optimal s'arrêtant au pixel (i, j) .



Si on continue la construction de ce graphe, celui-ci croît exponentiellement jusqu'à rencontrer un cas de base. Cette implémentation n'est pas optimale, les appels récursifs étant exponentiels et des résultats obtenus précédemment sont recalculés. Une approche par programmation dynamique faisant intervenir la mémorisation de certains coûts réduirait le temps de calcul.

3.4 Pseudo-code du calcul du coût du sillon optimal

La fonction `COMPUTEOPTIMALGROOVECOST` prend en argument une table des coûts dont l'élément (i, j) renseigne sur le coût du pixel (i, j) .

`COMPUTEOPTIMALGROOVECOST(cTable)`

3.5 Pseudo-code pour renvoyer l'image réduite de k pixels en largeur

3.6 Complexité en temps et en espace