Université de Rennes 1 Année 2013-2014
ESIR2-INC

TP : Seam carving

1 Données Fournies

Les données relatives à ce TP se trouvent dans le répertoire partagé :
`/share/esir2/tsi/tpSeamCarver`
Vous y trouverez les fichiers et sous-répertoires suivants :

- `./images` : des images de test au format pgm. Les images `loup.pgm` et `mer.pgm` sont pour le seam carving;
- `tpSeam.cpp` : squelette de programme que vous pourrez utiliser.
- `ImagesRW.cpp(.h)` : entrées, sorties PGM
- `Makefile` : fichier de compilation
- `mask1.pgm` et `mask2.pgm` : deux masques pour la partie *seam carving*.

2 Redimensionnement d'images par *seam carving*

L'algorithme de *seam carving* est présenté `http://www.youtube.com/watch?v=c-SSu3tJ3ns`.

2.1 Principe

Le *seam carving* est un algorithme de redimensionnement d'images par suppression de pixels de *moindre importance*. Il existe différentes méthodes pour choisir les pixels de *moindre importance*. Ici, nous utiliserons la mesure de gradient pour déterminer l'énergie d'un pixel.

Soit I une matrice de pixels de taille $l \times c$. On appelle couture (seam) horizontale de I, un ensemble de c pixels $\{p_k = (i_k, j_k),\ 0 \leq k \leq c-1\}$ dont les coordonnées (i_k, j_k) vérifient :

$$\begin{cases} j_k & = k \\ |i_{k+1} - i_k| & \leq 1. \end{cases} \tag{1}$$

De la même façon, une couture verticale de I est définie par un ensemble de l pixels $\{p_k = (i_k, j_k),\ 0 \leq k \leq l-1\}$ dont les coordonnées (i_k, j_k) vérifient :

$$\begin{cases} i_k & = k \\ |j_{k+1} - j_k| & \leq 1. \end{cases} \tag{2}$$

On définit l'énergie d'une couture par la somme des énergies des pixels la composants. L'objectif du redimensionnement d'image par seam carving est de retirer les coutures d'énergie minimale. On réduit ainsi le nombre de lignes d'une image en enlevant les coutures horizontales d'énergie minimale et le nombre de colonnes en enlevant les coutures verticales d'énergie minimale.

Vous procéderez en 3 étapes pour réaliser le seam carver:

1. Calcul de la matrice d'énergie de l'image
2. Détermination de la couture d'énergie minimale (algorithme de Viterbi)
3. Redimensionnement de l'image en retirant la couture

Il est conseillé de travailler uniquement avec des coutures horizontales. Vous pourrez transposer l'image pour travailler sur des coutures verticales.

Tester votre seam carver sur les images `loup.pgm` et `mer.pgm` ainsi que sur des images de votre choix qui refléteront les intérêts et les limites de ce procédé.

2.2 Zones d'intérêts

Il est possible de définir des zones dans l'image que l'on souhaite garder ou enlever en priorité. Pour cela, il suffit de modifier la matrice d'énergie de l'image avant le calcul des coutures d'énergie minimale. Si l'on souhaite enlever une zone en priorité, on peut mettre l'énergie de cette zone à 0 par exemple. Inversement, pour protéger une zone, on peut lui mettre des valeurs d'énergie très fortes.

Vous avez à votre disposition deux fichiers `mask1.pgm` et `mask2.pgm` qui correspondent à deux zones de l'image `mer.pgm` : la deuxième personne en partant de la gauche, et l'espace entre la troisième et la quatrième personne en partant de la gauche.

Utilisez ces masques pour enlever en priorité la zone indiquée par `mask1.pgm` et protéger la zone indiquée par `mask2.pgm`.

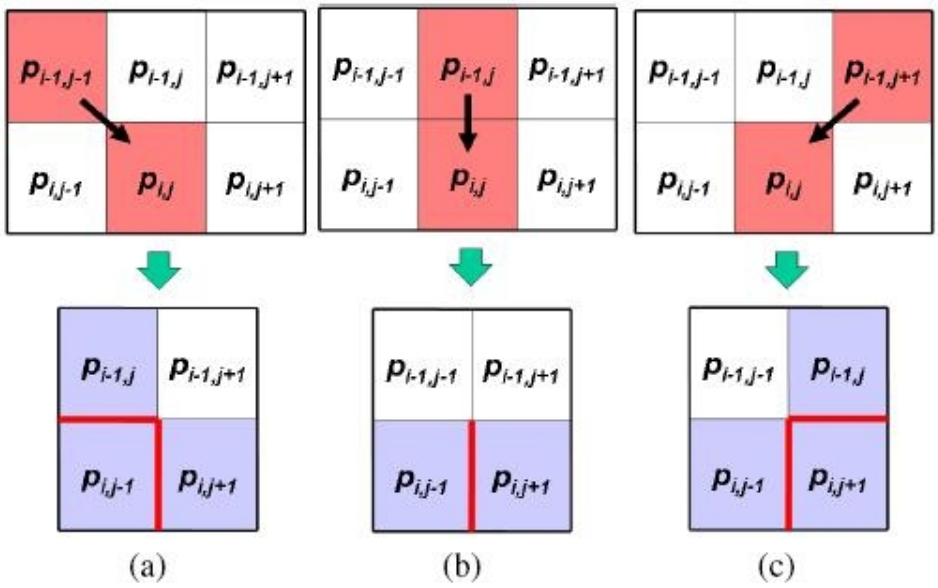
3 Modifications et améliorations

3.1 Fonction d'énergie

Dans les sections précédentes, la fonction d'énergie utilisée correspondait au gradient de l'image. Proposez une ou deux autres façons de calculer l'énergie. Vous évaluerez l'impact sur le redimensionnement des images.

3.2 Backward vs Forward énergie (BONUS)

Jusqu'à présent, l'agorithme développé utilise une énergie dite *Backward* (on traite l'image de bas en haut pour on remonte à partir de la couture ayant la plus faible énergie). Il existe une autre solution dite *Forward*. Cette solution mesure l'énergie **inserée** dans l'image après la suppression d'une couture. Cette énergie est due aux nouveaux contours créés (pixels qui n'étaient pas initialement proches mais qui le deviennent après la suppression de la couture). Cette énergie se mesure donc avec une différence, dite *Forward*, entre les pixels qui deviennent voisins. Trois cas peuvent être considérés comme indiqué par la figure qui suit:



L'algorithme est le suivant:

$$M(i,j) = S(i,j) + \min \begin{cases} M(i-1,j-1) + C_L(i,j) \\ M(i-1,j) + C_U(i,j) \\ M(i-1,j+1) + C_R(i,j) \end{cases} \tag{3}$$

avec,

$$\begin{cases} C_L(i,j) = |I(i,j+1) - I(i,j-1)| + |I(i-1,j) - I(i,j-1)| \\ C_U(i,j) = |I(i,j+1) - I(i,j-1)| \\ C_R(i,j) = |I(i,j+1) - I(i,j-1)| + |I(i-1,j) - I(i,j+1)| \end{cases} \tag{4}$$

Implémentez cette algorithme et comparez le résultat obtenu avec la version *Backward*.