

# Seam Carving

Bruno Guillon

*bruno.guillon@liafa.univ-paris-diderot.fr*

PI4 – Projet Informatique

## Résumé

Le but de ce projet est de réaliser un programme qui permet de redimensionner des images sans déformation, en supprimant des lignes de pixels “inutiles” selon la méthode du *Seam Carving*.

## 1 Description

Le Seamcarving est un algorithme de réduction d’image, dont l’objectif est de garder les proportions de l’objet tout en supprimant des informations peu utiles de l’image. Une démonstration vaut mieux qu’un long discours, allez donc voir cette vidéo (en anglais). Je vous invite également à aller voir la page wikipedia.

### 1.1 Image

Une image est une matrice rectangulaire dont les coefficients sont des triplets (r,g,b) (pour red (rouge), green (vert) et blue (bleu)). L’idée principale du Seam Carving est de supprimer une *ligne* verticale (ou horizontale) de pixels *inutiles*.

### 1.2 Énergie

L’utilité d’un pixel est donnée par son *énergie* qui est une fonction de son voisinage : s’il a une couleur (r,g,b) proche de ses voisins, il a une faible énergie, au contraire, s’il est très différents de ses voisins, son énergie sera forte. Dans l’idée, un pixel bleu au milieu d’une mer bleu est moins important que le pixel rouge (d’une bouée par exemple) à côté des pixels bleu de la mer. Il vous faudra donc calculer et stocker l’énergie de chaque pixel.

### 1.3 Lignes d’énergies minimales

À partir de cette énergie, vous devrez trouver les lignes verticales ou horizontales, d’énergie minimale, que vous supprimerez. Une ligne verticale est une liste de coordonnées  $(i_0, j_0), (i_1, j_1), \dots, (i_{h-1}, j_{h-1})$  où  $h$  est la hauteur de l’image et pour tout  $0 < k < h$ , on ait  $|i_{k+1} - i_k| \leq 1$  et pour tout  $k$ ,  $j_k = k$ .

Autrement dit, c'est une suite de point voisins, qui traverse l'image de haut en bas (un point par ligne de pixel).

## 2 Projet

Le projet sera un programme capable de lire les images au format bmp (non compressées) et de redimensionner l'image à l'aide du Seam Carving.

Le programme devra intégrer une interface graphique, permettant de diminuer manuellement les dimensions verticales et horizontales d'une image affichée. Ceci devra être fait de deux façons :

1. cliquer-glisser (sur un bord ou un coin) pour redimensionner l'image (ce qui impose d'être très efficace, on pourra éventuellement se passer de "l'affichage en live", mais ce serait mieux avec) ;
2. entrer directement la nouvelle taille d'image souhaitée.

L'interface devra également inclure une fonction d'annulation qui permettra d'annuler **les** dernières opérations effectuées par l'utilisateur.

Vous devrez ensuite vous intéresser à l'extension des images qui fonctionne selon un principe similaire (ajouter des pixels similaires à leur voisins, dans des zones de faible énergie).

Si l'utilisateur diminue puis étend une image (dans la même session, bien sûr), votre programme devra retrouver les pixels précédemment effacés pour les ajouter, tant que c'est possible.

Votre programme devra permettre la lecture et l'écriture (sauvegarde) d'image.

## 3 Pour aller plus loin

Vous devez implanter au moins une des extensions suivantes :

1. Gérer plusieurs formats d'image (jpg, png, gif)...
2. Permettre à l'utilisateur de définir des zones d'énergie très faible ou très forte (cf. vidéo), pour avoir plus de contrôle sur la suppression.
3. (\*) Permettre à l'utilisateur de signaler des alignements, qui devront être préservés dans l'image résultat : par exemple, si un fil électrique traverse en oblique l'image de gauche à droite, on veut qu'il reste continu et approximativement droit, après une série de suppression (une piste serait d'augmenter l'énergie d'un pixel sur cette ligne lorsqu'un de ses voisins sur la ligne a été supprimé).
4. Une autre extension intéressante à discuter avec votre chargé de TP.