

Projet Caml: Seam Carving

Jean-Marie Madiot, Sylvain Dailler

20 Novembre 2008

Résumé

Le but de ce projet est d'écrire un programme Caml permettant de lire une image et de la redimensionner par la méthode du seam carving.

1 Cahier des Charges

Réalisation d'un programme permettant de :

- ouvrir une image en couleur de format quelconque
- utiliser une fonction d'énergie afin de redimensionner cette image
- afficher cette image
- redimensionner l'image en hauteur ou/et en largeur
- permettre la modification de la fonction d'énergie par la souris
- enregistrer l'image dans un fichier

2 Utilisation

```
./seamcarving image.jpg
```

On peut préciser le fichier de sortie, le nom par défaut étant `output.ppm`.

```
./seamcarving image.jpg resultat.jpg
```

L'image de sortie sera stockée dans un fichier du nom indiqué, en respectant le format indiqué. (C'est la magie de `convert` qui opère.)

3 Fichiers et modules utilisés

Nous avons utilisé plusieurs fichiers pour réaliser ce projet :

- `interface.ml` permet de gérer l'ensemble de l'interface
- `ppm.ml` permet de gérer les images. Il permet de gérer des fonctions de lecture d'images en ppm ainsi que de les convertir en matrice.
- `main.ml` est le fichier principal qui utilise les fonctions définies dans les autres fichiers.
- `traitement.ml` permet de gérer les opérations sur les matrices d'images (enlever une colonne).

4 Solutions utilisées et choix effectués

4.1 Ouverture d'images

On gère à l'aide d'ImageMagick (`convert`) un grand nombre de formats d'image. Le programme lit une image PPM version P6, en lisant le format, puis la taille de l'image, puis les pixels de l'image octet par octet (fichier binaire, et non texte). La donnée de l'image est donc stockée dans une matrice de triplets (rouge, vert, bleu).

4.2 Traitement de l'image

Principe : on détermine pour chaque pixel son importance, appelée énergie du pixel, grâce à une fonction d'énergie, dépendant notamment de la variation de l'intensité de chaque composante autour du pixel.

4.2.1 Fonction d'énergie

Pour une image en noir et blanc (niveaux de gris), la fonction d'énergie donnée $\sqrt{(d/dx)^2 + (d/dy)^2}$ fonctionne, mais il a fallu l'améliorer pour une image en couleurs.

On a implémenté la fonction d'énergie utilisant le gradient. La fonction utilise les trois composantes RGB de chaque pixel :

$$\begin{aligned}dR_{i,j} &= (R_{i,j+1} - R_{i,j-1})^2 + (R_{i+1,j} - R_{i-1,j})^2 \\dG_{i,j} &= (G_{i,j+1} - G_{i,j-1})^2 + (G_{i+1,j} - G_{i-1,j})^2 \\dB_{i,j} &= (B_{i,j+1} - B_{i,j-1})^2 + (B_{i+1,j} - B_{i-1,j})^2\end{aligned}$$

On choisit alors :

$$energie_{i,j} = \sqrt{dR_{i,j} + dG_{i,j} + dB_{i,j}}$$

De nombreuses fonctions d'énergies sont possibles.

Cette fonction d'énergie pose un problème aux bords car on ne peut pas utiliser la dérivée. La formule aux bords devient donc (exemple bord gauche) :

$$dR_{i,0} = (R_{i,1} - R_{i,0})^2 + (R_{i+1,0} - R_{i-1,0})^2$$

4.2.2 Destruction d'une colonne

On stocke la colonne à enlever dans un tableau. Puis, on crée une autre matrice en copiant la matrice initiale dans cette matrice sans la colonne à enlever. On peut appliquer cette fonction à la fois à l'image et à la matrice d'énergie pour garder la correspondance.

Le choix de conserver toutes les valeurs de la matrice d'énergie en supprimant les mêmes pixels sans recalcul est totalement réfléchi : empiriquement, les résultats sont tout à fait concluants. Cette méthode permet de conserver les choix de l'utilisateur qui est roi.

4.3 Interface graphique

L'utilisateur peut déterminer à l'aide de la souris la taille de l'image finale, et modifier de même la matrice d'énergie. On utilise à profit le module Graphics de Caml, notamment la synchronisation manuelle pour éviter tout type de scintillement.

On a créé le module `Interface` qui permet de regrouper les fonctions d'interface. La modularité accélère le développement des interfaces car

l'objectif est fixé, et la manière dont l'utilisateur entre les données n'intervient pas, permettant ainsi au programmeur d'interface d'effectuer des modifications sans interférer avec les programmeurs utilisant le module. (C'est d'autant plus vrai pour les interfaces que le développement d'une nouvelle fonctionnalité implique souvent de réfléchir à une interface correspondante.)

5 Perspectives d'évolution

On pourrait trouver d'autres fonctions d'énergie à implémenter.
On pourrait ajouter un mode pour agrandir les images, aussi.