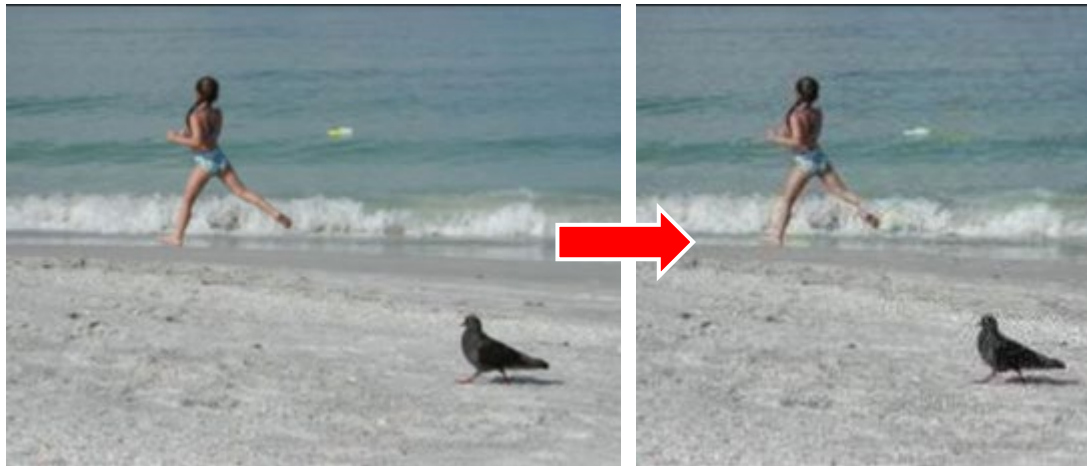


# Seam Carving

Compte rendu TP1 Vision Robotique



WU Zijian

([zijian-wu@outlook.com](mailto:zijian-wu@outlook.com))

## Répertoire

1	Introduction.....	2
2	Codage .....	2
3	Résultat .....	3
3.1	Banc.....	4
3.2	Plage .....	5
3.3	Oiseau .....	5
4	Conclusion .....	5
5	Référence .....	6

# 1 Introduction

J'ai terminé ce TP indépendamment pour une raison quelconque.

Seam Carving est un algorithme qui permet de couper l'image adaptative basée sur le principe de l'énergie. En termes simples, il est possible de conserver autant que possible des objets importants dans l'image, sans la déformer, et de couper ces objets avec moins d'importance.

Le principe de l'algorithme consiste à utiliser le gradient en tant qu'énergie. Le principe de flux maximal permet de diviser l'image en deux parties, le flux maximal trouvé correspondant à la position à couper.

L'image ci-dessous est le processus que j'ai conçu.

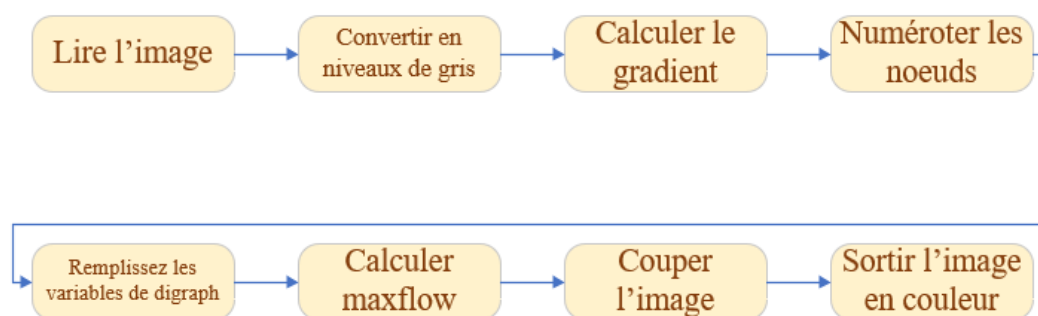


Figure 1 Processus

Il y a plusieurs façons de calculer l'énergie. J'ai choisi une méthode dans la référence, comme le montre la figure 2. Cette méthode considère uniquement la relation de gradient entre chaque pixel et le pixel adjacent à son côté droit, quel que soit le gradient vertical, et ne permet pas une recherche inversée d'énergie, ce qui signifie que l'énergie de position "infini" sur la figure est suffisamment grand.

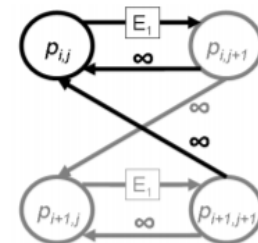


Figure 2 Energie

## 2 Codage

J'ai d'abord testé si l'algorithme est réalisable. J'ai créé une image  $5 \times 5$ , comme illustré à la figure 3, puis numéroté chaque nœud dans le diagramme. Je place le nœud 1 à l'extrême gauche, le nœud 2 à l'extrême droite, puis chaque pixel du graphique est un nœud, trié à partir du 3ème.

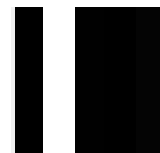


Figure 3 Image testé

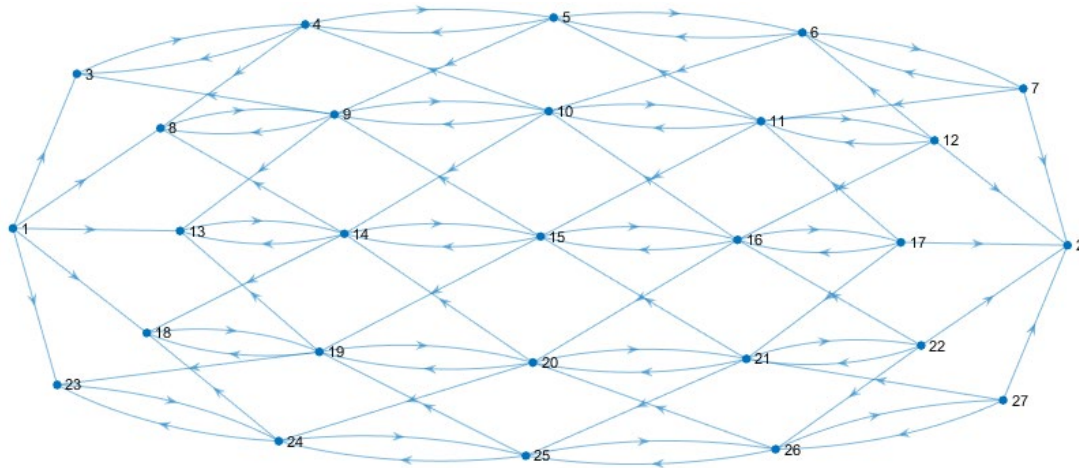


Figure 4 Image des nœuds

La figure ci-dessus montre la numérotation de mes nœuds. Le premier pixel de la première ligne est numéroté 3, puis incrémenté. De plus, je connecte le nœud s (numéro 1) à tous les nœuds de la première colonne et connecte tous les nœuds de la dernière colonne au nœud t (numéro 2). Ensuite, j'utilise la fonction maxflow pour calculer la position du plus grand flux et des éléments contenus dans S et T.

```
>> t'
ans =
    2    5    6    7   10   11   12   15   16   17   20   21   22   24   25   26   27

>> s'
ans =
    1    3    4    8    9   13   14   18   19   23
```

Figure 5 Résultat de maxflow

De plus, j'ai conçu l'algorithme de recadrage et l'algorithme de redessinage de l'image. La figure 'cut' montre la position de la coupe, qui est dessinée en rouge sur la figure. Cette partie a été complétée après ma discussion avec mon camarade Boyang.

### 3 Résultat



Figure 6 programme en cours d'exécution

Dans mon algorithme, la limite d'écrtage de l'image est limitée aux carrés. Cela signifie que si la longueur de l'image n'est pas supérieure à la largeur, le découpage est arrêté. Tous mes résultats sont donc affichés sous forme d'images carrées. L'image de gauche est une image générée par un programme en cours d'exécution. Vous pouvez voir la ligne rouge indiquant la pièce en cours de traitement. Dans le programme, nous pouvons définir manuellement la taille de l'étape, qui correspond au nombre de pixels ajustés à la fois. J'ai essayé de traiter 3 pixels à la fois et seulement 1 pixel à la fois, et l'impact final sur les résultats n'est pas important. Ensuite, je vais analyser les trois images une par une.

### 3.1 Banc



Figure 7 Image original



Figure 8 Image coupé

L'effet de recadrage de cette image n'est pas idéal, car pour le banc, la partie de l'énergie la plus basse ne se trouve pas au centre de l'image. J'ai constaté qu'au cours du processus de coupage, la première partie de la coupe se situait aux extrémités gauche et droite de la photo, car la zone sans banc avait la plus faible énergie. Ensuite, la zone qui commence à être coupée est la partie centrale de la photo, suivie de l'extrémité gauche du banc. [Cliquez ici pour le processus de personnalisation.](#)

Le banc sur la photo est incomplet. Je pensais au départ que c'était parce que la taille de l'étape était définie sur 3 pixels, ce qui signifie que chaque fois que maxflow est trouvé, les trois pixels autour de maxflow sont coupés. Mais après avoir essayé de changer la taille du pas à 1 pixel et de l'avoir répété plus de 300 fois, j'ai trouvé que le résultat était presque identique. Figure 10 montre une taille de pas de 1 pixel.

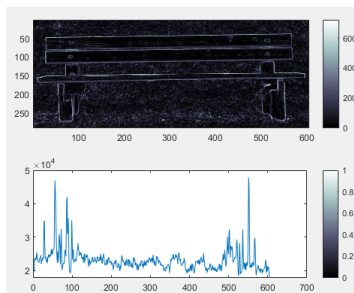


Figure 9 Image d'Energie



Figure 10 Image coupé avec pas=1px

J'ai calculé l'énergie et superposé l'énergie de chaque colonne pour obtenir la figure 9. Grâce à l'observation, j'ai constaté que l'énergie du pied sur le côté gauche du banc est effectivement plus petite que la partie centrale de l'image. Cela explique pourquoi le côté gauche du banc a été coupé. Afin de résoudre ce problème, je pense que le seul moyen réalisable est de modifier la méthode de calcul de l'énergie. Je pense que nous devons prendre en compte la relation de gradient entre chaque pixel et ses pixels supérieurs et inférieurs lors du calcul de l'énergie. C'est la pratique de la méthode d dans la référence, Le gradient longitudinal est calculé par  $+ -LU$ . Cependant, je n'ai pas essayé cela.

## 3.2 Plage



Figure 11 Image original



Figure 12 Image coupé

Pour cette photo, la coupe est meilleure. Le processus de coupe préserve complètement les gens et les pigeons sur la photo. Cette image est de petite taille et fonctionne rapidement. Je sors une image au format GIF pour voir visuellement le traitement. ([Cliquez ici](#))

## 3.3 Oiseau



Figure 13 Image original



Figure 14 Image coupé

Cette image est de très grande taille et nécessite beaucoup de pixels pour être traitée. J'ai donc changé la taille du pas à 10 pixels et chaque itération a supprimé 10 pixels autour de maxflow. Nous pouvons voir que les oiseaux sont bien conservés et que les branches et les feuilles sont comprimées. Surtout la feuille dans le coin inférieur gauche, un seul bord est laissé après avoir été coupé. Cela est dû au fait que le gradient longitudinal n'est pas pris en compte dans le calcul de l'énergie. Étant donné que le premier pixel de cette ligne est vert et que le gradient suivant est petit, le pixel est coupé, même si le gradient du pixel est élevé avec les pixels au-dessus. La taille de cette image étant très grande, le programme dure une nuit pour obtenir le résultat. Je ne mettrai donc pas l'image gif.

# 4 Conclusion

Le but de ce TP est de comprendre la signification de maxflow et d'utiliser maxflow pour le coupage de Seam Carving. Je pense que je comprends bien la signification de maxflow et

que j'atteins le but, même s'il y a des imperfections dans le code.

Au début, je ne comprenais pas le maxflow nécessaire pour trouver l'ensemble de la photo ; j'ai donc essayé de trouver leur maxflow tous les quatre pixels, puis j'ai essayé d'ajouter ces mfs pour calculer la valeur finale de mf. Jusqu'à ce que je rentre à la maison, je me suis rendu compte que je pouvais utiliser la fonction Matlab pour trouver le mf de l'image en une fois.

La difficulté avec ce TP est que chaque pixel est numéroté régulièrement. Bien que nous puissions numéroter consécutivement, nous devons trouver la règle de numérotation pour chaque pixel et le pixel situé en dessous.

De plus, cet algorithme n'est pas difficile à écrire, mais il est difficile à comprendre. De plus, il est plus difficile d'obtenir les images divisées de S et T obtenues de maxflow et de les extraire.

## 5 Référence

<http://www.faculty.idc.ac.il/arik/SCWeb/vidret/vidret.pdf>