

3D Computer Vision

Graph Cuts: Estimation de la carte de disparité

Mohammed Amine KHELDOUNI

5 novembre 2018

1 *Graph cut* et carte de disparité

L'objectif de cet exercice est, comme celui du précédent travail, d'estimer la carte de disparité d'une image étant donné deux perspectives différentes d'un élément photographié. Lors du précédent rendu, nous avons effectué un algorithme de propagation de graines (*seeds*), qui propage les correspondances les plus corrélées (ayant un $NCC \geq 95\%$) sur les 4 pixels voisins. Cette fois-ci, on se propose de résoudre le calcul de la disparité par un algorithme de *graph cut*.

Pour cela, nous considérons un intervalle de nombres entiers, et on cherche pour chaque pixel, la disparité $d \in \{d_{min} = 10, \dots, d_{max} = 55\}$ le liant à son homologue de la second image. La construction du graphe se fait en modélisant les pixels de l'image par des noeuds et les différentes valeurs possibles de disparité $h = d_{max} - d_{min} + 1$ comme étant des couches (*layers*). Enfin, on attribue les poids des arêtes du graphe selon la formule définie dans le cours :

$$D(x, y) = w_{cc} \rho(ZNCC(I_1(x, y), I_2(x + d, y), n) + 1 + 4\lambda$$

où n est le rayon du patch considéré. Nous verrons dans la prochaine section en quoi la régularisation est importante par rapport au poids des arêtes.

2 Illustrations et comparaison des deux modèles

Dans cet exemple, nous considérons les deux images ci-dessous, sur lesquelles nous allons estimé une carte de disparité par les deux méthodes, algorithme de propagation de graines et par un algorithme de *graph cut*.

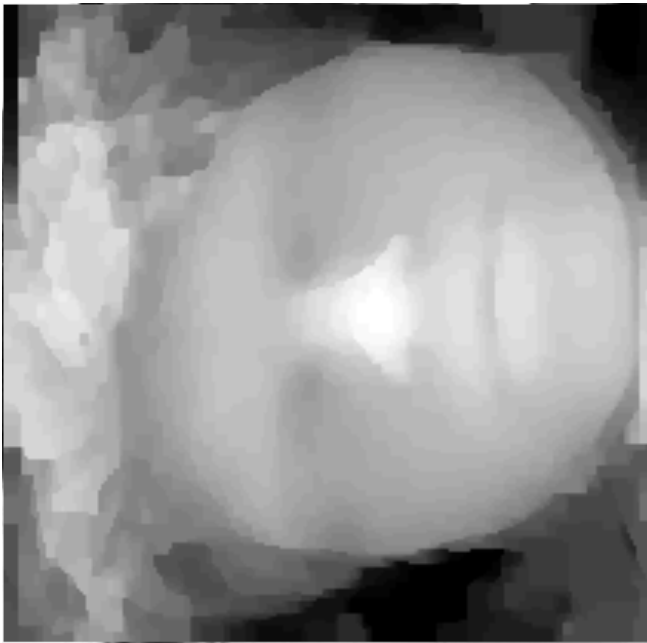


Input image avec la tête légèrement vers l'arrière

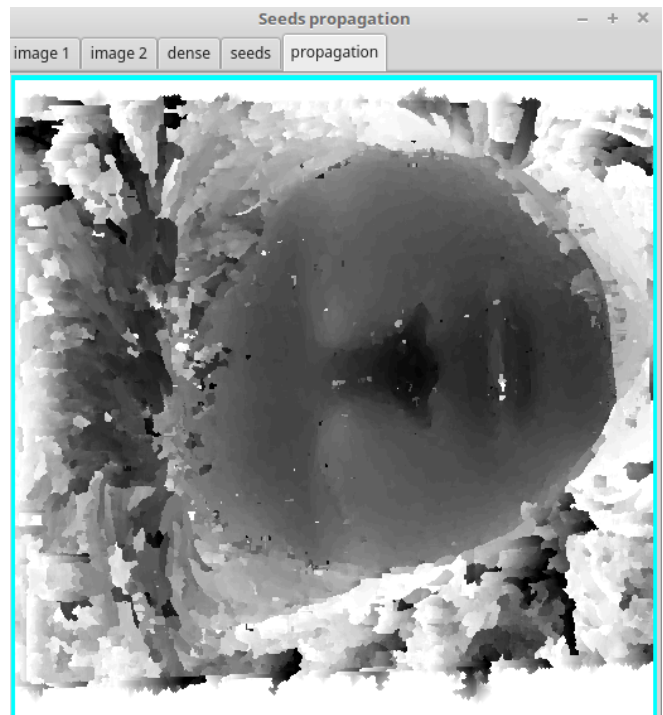


Input image avec la tête droite

Les deux algorithmes utilisés estiment la carte de disparité sur un ensemble de nombres entiers ($d \in \{d_{min} = 10, \dots, d_{max} = 55\}$). Les cartes de disparités des deux algorithmes sont fournies ci-dessous.



Carte de disparité estimée par un algorithme de *Graph Cut*



Au niveau de la visualisation 3D des deux méthodes, on remarque que la méthode du *Graph Cut* est supérieure grâce au terme de régularisation qu'elle emploie dans le calcul de son coût. Les figures ci-dessous représentent le modèle 3D de chacune des deux méthodes.

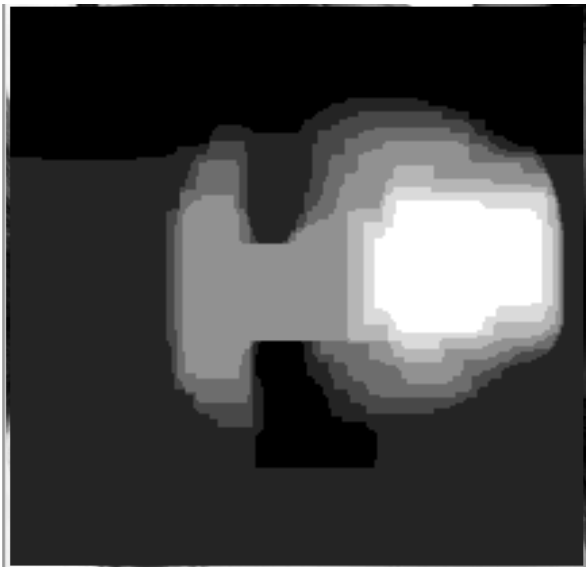


Visualisation de l'image en 3D après estimation de la disparité par *Graph Cut*

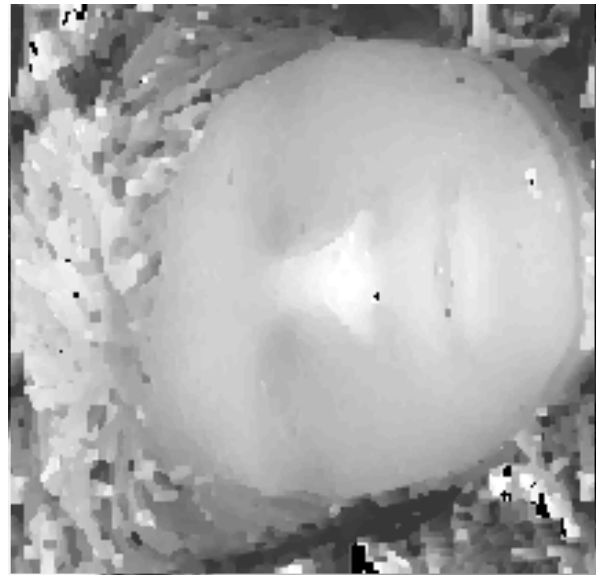


Visualisation de l'image en 3D après estimation de la disparité par propagation de *seeds*

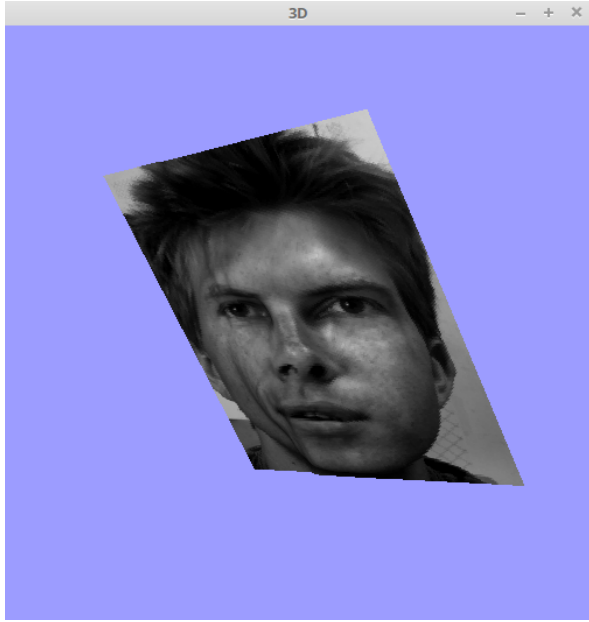
Pour évaluer l'importance de la régularisation et du paramétrage en λ_f du modèle de *Graph Cut*, nous illustrons ici un exemple avec un grand poids sur la régularisation ($\lambda_f = 0.9$) et un autre avec une régularisation très faible ($\lambda_f = 0.01$).



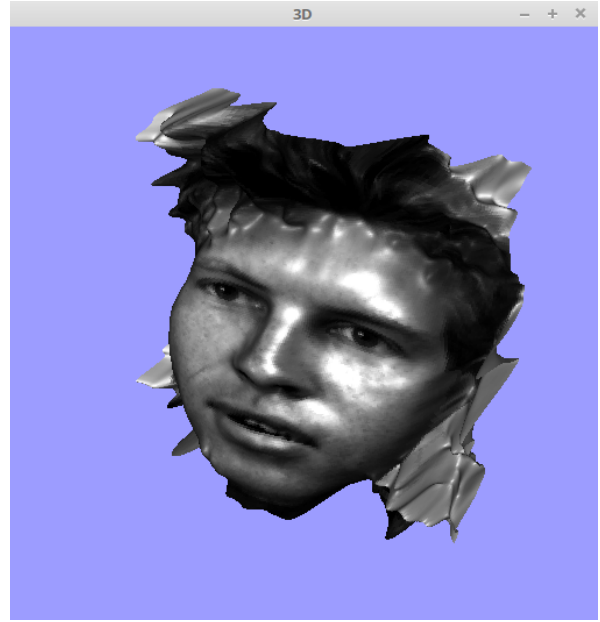
Visualisation de la disparité avec une forte régularisation ($\lambda_f = 0.9$)



Visualisation de la disparité avec une faible régularisation ($\lambda_f = 0.01$)



Visualisation de l'image en 3D avec une forte régularisation ($\lambda_f = 0.9$)



Visualisation de l'image en 3D avec une faible régularisation ($\lambda_f = 0.01$)

On remarque qu'une trop grande régularisation empêche le graphe d'optimiser les calculs par rapport à la corrélation croisée et qu'une faible régularisation (voire inexistante) entraîne un *overfitting* sur la figure qui se calibre parfaitement aux valeurs optimales mais ne parvient pas à aboutir à un modèle 3D lisse et fidèle aux attentes.

3 Conclusion

Pour conclure, nous avons bien réussi à illustrer deux exemples d'estimation du carte de disparité à partir de deux images. Nous avons remarqué la pertinence de l'algorithme de *Graph Cut* qui permet d'inclure un terme de régularisation qui contre balance les défauts d'un *overfitting* pour lisser l'image et restituer une image 3D plus fidèle à la réalité.