

Disparity map computation using Graph-cut method
TP #5
3D Computer Vision

Alexandre THIS

November 12, 2014

Contents

1	Résultats obtenus	2
2	Comparaison avec la méthode de propagation de graine	3
3	Binarisation et segmentation	4

1 Résultats obtenus

Les images d'origines sont les suivantes :



Figure 1: Images d'origine

La carte de disparité calculée avec la méthode Graph cut est la suivante :

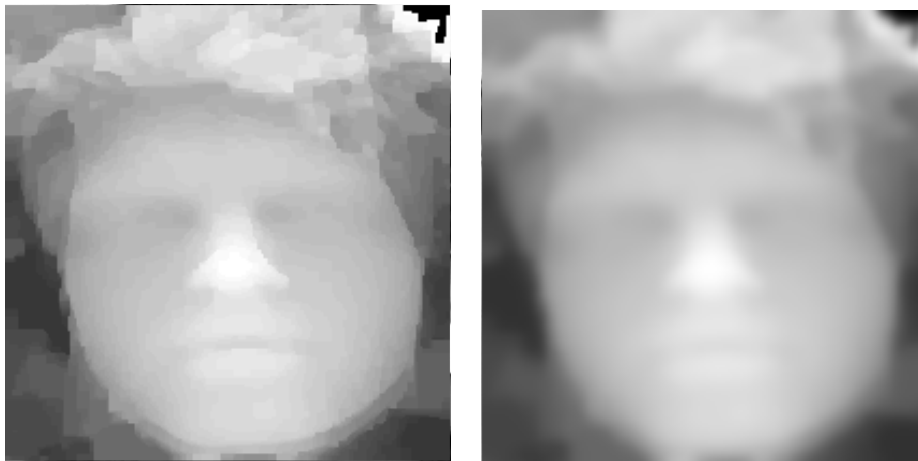


Figure 2: Carte de disparité obtenue par Graph-cut

La carte de disparité a permis d'obtenir le mesh 3D suivant :

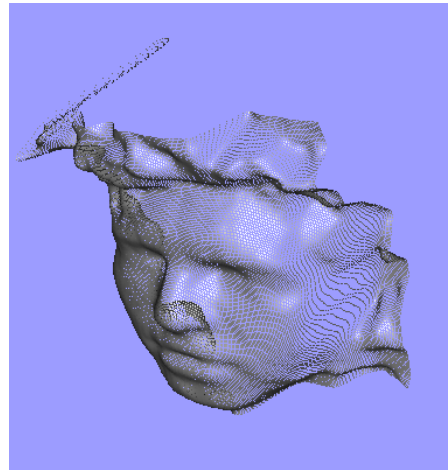


Figure 3: Mesh de la carte de disparité obtenue par Graph-cut

2 Comparaison avec la méthode de propagation de graine

On peut observer une nette différence avec la carte de disparité obtenue par propagation de graines :

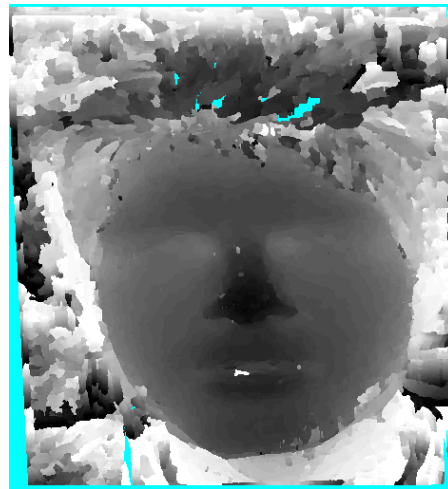


Figure 4: Gauche : Graph-cut, Droite : Propagation de graine

Malgré l'utilisation de paramètres différents (zoom notamment), on peut remarquer que le calcul de carte de disparité en utilisant la propagation de graine produit de forts gradients dans la carte de disparités. Le terme de régularisation de la méthode de graph cut permet de réduire ce genre de défauts.

La reconstruction 3D en utilisant la carte de disparité obtenue par propagation de graine est de la même façon moins "régulière" (il est important de noter que les méthodes d'affichage du mesh

ne sont pas les même pour les deux méthodes. La méthode utilisée pour cette image est celle fournie par Mr Monasse dans le TP4 :

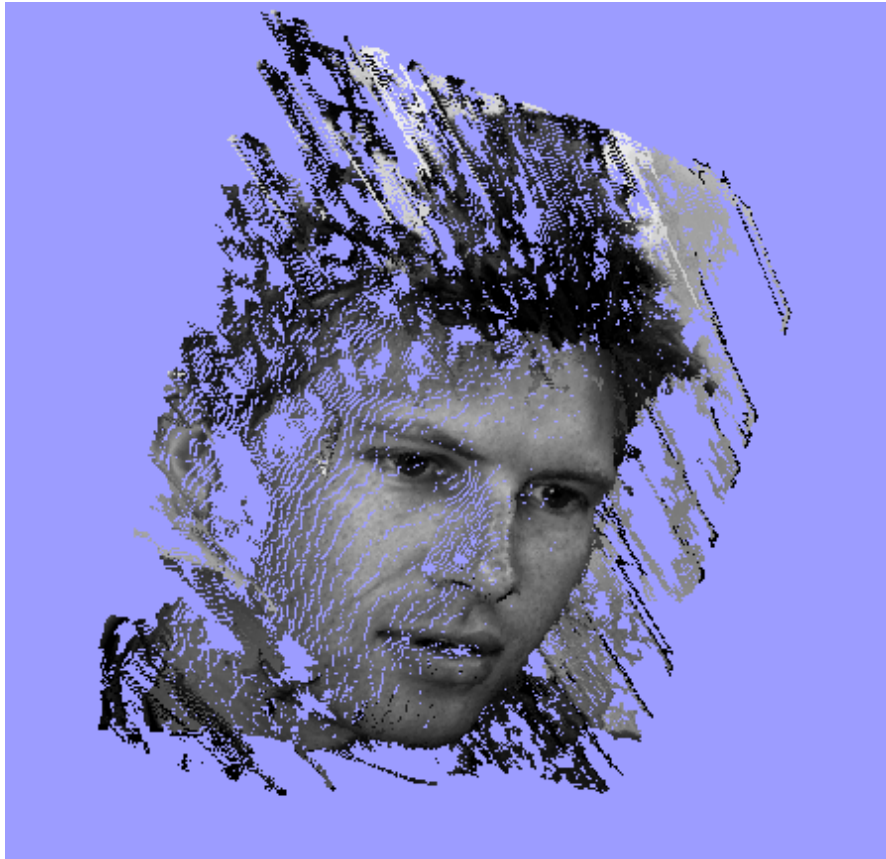


Figure 5: Mesh de la carte de disparité obtenue par propagation de graine

D'autre part, il est important de noter que le temps de calcul pour la méthode Graph cut est beaucoup plus important que pour la méthode de propagation de graine. En effet, même avec un facteur zoom égal à 2 pour le calcul par Graph cut, le temps de calcul sur ma machine est de l'ordre de 30 secondes. La méthode de propagation de graine, avec un facteur zoom égal à 1, est réalisée environ en 5 secondes.

3 Binarisation et segmentation

Les méthodes par graph cut permettent de réaliser la binarisation et la segmentation d'image. Bien que ce n'est pas le sujet de ce TP, les résultats suivants ont été obtenus (le code pour la binarisation et la segmentation se trouve dans le fichier `exampleGC.cpp`):



Figure 6: binarisation



Figure 7: Segmentation



Figure 8: Superposition de la segmentation et de l'image d'origine