

Rapport sur le TP 4

Cours de Vision 3D Artificielle – Master MVA

Matthieu Boyer

04 Novembre 2025

Présentation du Rapport

Ce rapport est en deux parties principales. Nous commencerons par une rapide présentation du code et de l'implémentation, puis étudierons quelques expériences qui ont été faites.

1 Implémentation

L'intégralité du code est disponible dans le fichier `GCDisparity.cpp` joint à ce rapport dans l'archive. Une fois compilé avec le fichier `CMake`, il possède deux modes principaux :

- Le mode normal, qui permet d'afficher la reconstruction 3D à partir des cartes de disparité générée par la méthode des coupes sur graphe (ci-après notée Graph Cuts ou GC) ;
- Le mode `test`, qui permet de générer les expériences ci-dessous.

Ces deux modes font principalement appel aux fonctions `build_graph` et `decode_graph` :

`build_graph` construit, à partir de la table des valeurs de NCC (co-corrélation normalisée) et d'un paramètre λ de régularisation, un graphe présentant un problème de flot adapté. Plus précisément, le coût depuis la source/vers le puits est la vraisemblance d'une valeur de disparité pour un pixel donné, et entre deux pixels, on met des arêtes selon les valeurs de NCC et le long du gradient de disparité.

`decode_graph` applique sur un graphe donné un algorithme permettant de résoudre le problème de flot minimum et permet donc de retrouver, au niveau de la cut trouvée dans le graphe, la séparation entre les sous-images ou la diparité est retenue et celles où la disparité est rejetée.

2 Expériences

Dans les tableaux de la Table 1 ci-dessous on regroupe, pour différentes valeurs du paramètre λ de régularisation et de la taille du voisinage utilisé pour l'algorithme NCC, plusieurs métriques permettant de mesurer la qualité des approximations obtenues en calculant par coupes maximales sur un graphe (méthode GC/Graph Cuts), par rapport à la méthode RG des régions grandissantes.

Ici, les métriques utilisées sont les suivantes :

- Le temps de calcul pour chaque méthode ;
- La moyenne des normes des gradients de variation ;
- L'erreur moyenne absolue, calculée entre les deux puisque la RG est prédit mieux les disparités.

Plus généralement, on trouve que la méthode GC donne des résultats légèrement moins précis que la méthode RG, et prend légèrement plus de temps, mais ceci vient en offrant des résultats bien plus lisses, ce qui permet une reconstruction visuellement plus réaliste. En moyenne on a les rapports de performance présentés dans la Table 2

TABLE 1 – Comparison of disparity estimation methods. Smoothness is measured as average gradient magnitude (lower is smoother). MAE shows mean absolute error between Graph Cuts and Region Growing results.

Method	Time (s)	Smooth.	Valid	MAE
GC	11.661	17.570	0.927	–
RG	6.239	17.228	1.000	16.779
(a) $\lambda = 0.050$, window radius = 1 (patch 3x3)				
Method	Time (s)	Smooth.	Valid	MAE
GC	35.300	8.995	0.928	–
RG	32.417	11.710	1.000	15.203
(c) $\lambda = 0.050$, window radius = 3 (patch 7x7)				
Method	Time (s)	Smooth.	Valid	MAE
GC	9.180	11.767	0.938	–
RG	6.240	17.228	1.000	16.914
(e) $\lambda = 0.100$, window radius = 1 (patch 3x3)				
Method	Time (s)	Smooth.	Valid	MAE
GC	35.473	5.917	0.942	–
RG	32.388	11.710	1.000	15.608
(g) $\lambda = 0.100$, window radius = 3 (patch 7x7)				
Method	Time (s)	Smooth.	Valid	MAE
GC	9.487	2.341	0.957	–
RG	6.338	17.228	1.000	17.672
(i) $\lambda = 0.200$, window radius = 1 (patch 3x3)				
Method	Time (s)	Smooth.	Valid	MAE
GC	36.458	2.073	0.958	–
RG	32.843	11.710	1.000	16.958
(k) $\lambda = 0.200$, window radius = 3 (patch 7x7)				
Method	Time (s)	Smooth.	Valid	MAE
GC	8.903	0.000	0.961	–
RG	6.323	17.228	1.000	18.364
(m) $\lambda = 0.500$, window radius = 1 (patch 3x3)				
Method	Time (s)	Smooth.	Valid	MAE
GC	36.020	0.074	0.960	–
RG	32.782	11.710	1.000	18.407
(o) $\lambda = 0.500$, window radius = 3 (patch 7x7)				

Method	Time (s)	Smooth.	Valid	MAE
GC	19.901	12.187	0.926	–
RG	17.090	14.125	1.000	16.019
(b) $\lambda = 0.050$, window radius = 2 (patch 5x5)				
Method	Time (s)	Smooth.	Valid	MAE
GC	56.491	7.044	0.928	–
RG	53.154	9.947	1.000	14.388
(d) $\lambda = 0.050$, window radius = 4 (patch 9x9)				
Method	Time (s)	Smooth.	Valid	MAE
GC	19.831	7.724	0.941	–
RG	16.794	14.125	1.000	16.301
(f) $\lambda = 0.100$, window radius = 2 (patch 5x5)				
Method	Time (s)	Smooth.	Valid	MAE
GC	56.278	4.707	0.943	–
RG	53.223	9.947	1.000	14.982
(h) $\lambda = 0.100$, window radius = 4 (patch 9x9)				
Method	Time (s)	Smooth.	Valid	MAE
GC	20.092	2.094	0.957	–
RG	16.781	14.125	1.000	17.293
(j) $\lambda = 0.200$, window radius = 2 (patch 5x5)				
Method	Time (s)	Smooth.	Valid	MAE
GC	57.824	1.977	0.958	–
RG	53.629	9.947	1.000	16.565
(l) $\lambda = 0.200$, window radius = 4 (patch 9x9)				
Method	Time (s)	Smooth.	Valid	MAE
GC	19.813	0.051	0.961	–
RG	17.043	14.125	1.000	18.397
(n) $\lambda = 0.500$, window radius = 2 (patch 5x5)				
Method	Time (s)	Smooth.	Valid	MAE
GC	56.661	0.104	0.960	–
RG	53.865	9.947	1.000	18.316
(p) $\lambda = 0.500$, window radius = 4 (patch 9x9)				

TABLE 2 – Rapports des valeurs de différentes métriques pour les deux méthodes. De gauche à droite : rapport des temps de calcul entre GC et RG (GC est x fois plus lent que RG) ; rapport (en % de la norme moyenne de GC) de la norme moyenne des gradients (GC produit des cartes $x\%$ plus lisses que RG) ; rapport des pourcentage d'exactitude (RG ayant toujours 100% d'exactitude, cela revient au pourcentage d'exactitude de GC).

Calc Time				
$r \backslash \lambda$	0.05	0.1	0.2	0.5
1	5.43	6.05	6.50	5.27
2	2.98	3.23	3.36	2.92
3	2.13	2.25	2.43	2.16
4	1.76	1.87	1.98	1.82

Smoothness				
$r \backslash \lambda$	0.05	0.1	0.2	0.5
1	0	31.7	86.4	100
2	13.7	45.3	85.2	99.6
3	23.2	49.5	82.3	99.4
4	29.2	52.7	80.1	99

Accuracy				
$r \backslash \lambda$	0.05	0.1	0.2	0.5
1	92.7	93.8	95.7	96.1
2	92.6	94.1	95.7	96.1
3	92.8	94.2	95.8	96
4	92.8	94.3	95.8	96