

Synthese

MVA		Note	TP1	TP2	TP3	TP4	Moy TP(/5)	Examen
ALKAN	Gaël	13.5	5	5	4.5	4.5	4.8	5
ANDREINI	Alexandre	14.5	4.5	4	4	4.5	4.3	10.5
AZIZIAN	Waïs	17.5	5	5	4.5	5	4.9	14.5
BATARDIERE	Bastien	12	4.5	5	3	4	4.1	5.5
BENADY	Antoine		3.5					
BERGER	Clément	16.5	4.5	4.5	5	5	4.8	12.5
COMBES	Theotime	12	5	5	3.5	4	4.4	4
DARCET	Thimothée	16	5	4.5	5	5	4.9	10.5
DOS SANTOS ROCHA	Danilo		5					
ER-RAMMACH	Ilyes	16	4.5	4.5	4.5	5	4.6	12
FERNANDEZ	Pierre	15	5	4.5	4.5	4.5	4.6	9.5
GALLAND	Lucie	17	4	5	5	4.5	4.6	15
GRAIVE	Martin	14.5	4.5	4	5	3.5	4.3	10.5
HAURET	Julien	10	3	2	4.5	4.5	3.5	4.5
JACOB	Paul	15.5	5	4.5	4.5	4.5	4.6	11
LEGER	Victor	13.5	3.5	5	4.5	3.5	4.1	9
LOTHE	Grégoire	15	5	3.5	5	4.5	4.5	10
MAAZOUN	Imen	13	4.5	4.5	5	4.5	4.6	5
MAISON	Lucas	13.5	5	5	4	4	4.5	7
MITTELMAN	Quentin	11	4	1.5	4.5	3.5	3.4	7.5
NGUYEN	Clément	11.5	3	5	5	3.5	4.1	3.5
ORIOLE	Benoît	14	4.5	4.5	4.5	4.5	4.5	8
REME	Raphael	13.5	5	5	5	5	5	4
RIOU	Auriane	15	3.5	4.5	5	4.5	4.4	11
SAUTIER	Corentin	14.5	4.5	5	5	4	4.6	8
SEGERIE	Charbel-Raphaël	16	4	4.5	4.5	5	4.5	12.5
SPINAT	Quentin	13.5	4	4	3.5	5	4.1	9
VANLAER	Youval	13	4	4	4.5	4.5	4.3	7

IMA		Note	TP1	TP2	TP3	TP4
ACKER	Edgar	8	2	4	2	0
ARGUELLO	Camillo	15.5	4.5	4	2.5	4.5
BOUDO	Louis	13.5	3	2.5	3.5	4.5
BOUMERDAS	Wassila	16.5	4	4	4.5	4
DI BON	Florian	14	4	1.5	4.5	4
DJERADA	Nacer	16	3	4.5	5	3.5
EL MADAFRI	Saad	18	4	4.5	5	4.5
FAURE	Maxime	13.5	3.5	4	2.5	3.5
FORTIER	Victor	17.5	5	3.5	4.5	4.5
GAVALDA	Louis	10.5	1	2	3.5	4
GERUSHTA	Valentyn	14.5	5	3	2	4.5
GUESMIA	Zoheir	16.5	4.5	4.5	3.5	4
HARROUZ	Wail	16	4	4	3.5	4.5
HUANG	Qijia	12.5	5	3.5	4	0
HUANG	Weiqin	15.5	4	3.5	3	5
LAZRAK	Mouad	4	0.5	1	0.5	2
LEMESLE-WELTI	Alban	14	4	2.5	3.5	4

Synthese

MAHTAL	Ryad	12.5	0	3.5	5	4
NDOKO	Arthur	16.5	5	3.5	3.5	4.5
RAMOUL	Samy Rayan	12.5	2	2	4	4.5
SAULOU	Thomas	14	4	2	4	4
SZYMCZAK	Alan	15.5	4	4	3.5	4
ZHANG	Shihao	14	5	4.5	Plagiat	4.5

Auditeurs libre		Note	TP1	TP2	TP3	TP4	Moy TP(/5)	Examen
MOLITOR	Pauline	10	4	1.5	4.5	2	3	5.5

Panorama

Panorama(/5)

MVA

ALKAN	Gaël	5	Travail excellent !
ANDREINI	Alexandre	4.5	Le fait de calculer inverse(H) pour chaque pixel est coûteux en temps de calcul. Les artefacts de couleur sur la zone superposée sont dus à un overflow des unsigned char dans Color+Color.
AZIZIAN	Waïs	5	TB
BATARDIERE	Bastien	4.5	Avec des anyGetMouse, la procédure de clics aurait pu être bien plus souple. Attention, les x=width et y=height sont en-dehors de l'image.
BENADY	Antoine	3.5	Les résultats ne sont pas très bons car tu oublies de diviser le resultat de H_1 v par sa 3eme composante. Détecter le recouvrement grâce à la couleur blanche du fond n'est pas idéal.
BERGER	Clément	4.5	Il manque juste la transparence dans la zone commune des deux images.
COMBES	Theotime	5	TB
DARCET	Thimothée	5	TB, mais il vaut mieux calculer l'inverse de H en dehors des boucles pour etre plus rapide, et diviser par 2 pour remultiplier par 2 les couleurs n'est pas très judicieux.
DOS SANTOS ROCHA	Danilo	5	TB
ER-RAMMACH	Ilyes	4.5	Il manque juste la transparence dans la zone commune des deux images.
FERNANDEZ	Pierre	5	TB
GALLAND	Lucie	4	Code très propre. Les coefficients non initialisés de la matrice A ne sont pas forcément à zéro, il faut le faire explicitement (par exemple, A.fill(0)). Il y a une distorsion des couleurs dans la transparence a cause d'un overflow des unsigned char codant les canaux.
GRAIVE	Martin	4.5	Il vaut mieux faire un pull plutôt qu'un push et essayer de masquer les blancs.
HAURET	Julien	3	Le processus de clic est trompeur: il attribue alternativement à pts1 et pts2 independamment de la fenetre cliquee. Il n'y a pas de raison en general que le pixel (x,y) de I soit aussi celui de I1 (meme si c'est le cas avec ces images). Tu fais en fait un push de I2 sur I1, pas un pull.
JACOB	Paul	5	TB
LEGER	Victor	3.5	Ca marche dans ce cas car l'image de gauche est bien en haut à gauche dans le panorama final. Cependant, le coin haut-droite de l'image 0007 est tronqué dans le panorama, alors qu'il y a du blanc en bas. C'est dû au fait que (x0,y0), offset logique dans l'image résultat, n'est pas pris en compte. Il manque aussi la transparence.
LOTHE	Grégoire	5	TB, mais compter sur la couleur blanche pour detecter le recouvrement n'est pas idéal.
MAAZOUN	Imen	4.5	Les distorsions de couleur sont dues à un overflow des unsigned char dans Color+Color.
MAISON	Lucas	5	TB

Panorama

MITTELMAN	Quentin	4	Il manque la transparence dans la zone commune. A noter que les coordonnées (0,0) sont admissibles dans une image, donc $c[0]>0$ devrait être $c[0]\geq 0$.
NGUYEN	Clément	3	Je ne comprends pas pourquoi tu fais à la fois un push et un pull de l1. De plus, tu oublies de diviser le resultat de H_{inv} v par sa 3eme composante dans le pull.
ORIOLE	Benoît	4.5	Resoudre le systeme $Hx=b$ pour chaque pixel est bien plus lent que calculer l'inverse de H une fois pour toutes et faire une multiplication matrice-vecteur.
REME	Raphael	5	TB
RIOU	Auriane	3.5	La distorsion des couleurs est due à un overflow des unsigned char dans $Color+Color$. Le recalage est mieux que ça déjà avec 4 points, il y a un bug dans la fonction panorama: $pixel/pixel[2]$ au lieu de $pixel/=pixel[2]$!
SAUTIER	Corentin	4.5	Il y a un tout petit bug, sur lequel je suis tombé par manque de chance : ligne 169; $y<=h$ devrait être $y<h$.
SEGERIE	Charbel-Rapha	4	La procédure de clic est un peu fastidieuse (on ne peut pas cliquer dans l'ordre qu'on veut entre images). Il est dommage de diviser par 2 les couleurs des parties non superposées. Défauts mineurs: on compte sur l'overflow dans $WHITE=(255)+couleur/2$, le pixel (0,0) est licite.
SPINAT	Quentin	4	Pas de transparence dans la zone commune. Les artefacts sont dus au choix du "push", moins bon que le "pull".
VANLAER	Youval	4	Tu as fait un push au lieu d'un pull, ce qui explique les artefacts. Recourir à une couleur spécifique (le blanc) pour détecter la superposition n'est pas une très bonne idée car l2 peut contenir de tels pixels.
IMA			
ACKER	Edgar	2	(devoir 2019) Plusieurs bugs qui empêchent le programme de fonctionner correctement. Il y a un signe moins superflu dans B, $A(2i,1)$ devrait être $y1$ et non $x1$, dans le panorama il faut diviser par $v[2]$ pour obtenir x et y...
ARGUELLO	Camillo	4.5	Bon ensemble. Il est dommage que la partie non commune de l'image 1 apparaisse pâle.
BOUDO	Louis	3	Les coordonnées logiques dans l'image panorama sont décalées de $x0$ et $y0$. Après application de Xi , il faut diviser par la 3ème composante pour obtenir les coordonnées. La vérification qu'un point est dans l'image n'est pas assez rigoureuse.
BOUMERDAS	Wassila	4	La transparence ne correspond pas à un max, mais à une moyenne. Il est dommage de recalculer l'inverse de H pour chaque pixel, ça ralentit le programme.
DI BON	Florian	4	Recalculer l'inverse de H pour chaque pixel est peu judicieux. Il manque la transparence sur la zone commune du panorama.

Panorama

DJERADA	Nacer	3	La procédure de clic est quasi inutilisable, car <code>getMouse</code> concerne la fenêtre active, qui doit changer après chaque clic. Les coordonnées logiques dans l'image panorama doivent être décalées de (x0,y0).
EL MADAFRI	Saad	4	Il y a un bug ligne 78, avec un signe – superflu. Ca provoque un mauvais recalage. Dommage car tout le reste est bon.
FAURE	Maxime	3.5	Si tu avais fait un “pull”, tu n’aurais pas eu à te donner tant de mal à cacher les défauts. Le code du panorama est trop complexe.
FORTIER	Victor	5	TB
GAVALDA	Louis	1	Le remplissage de la matrice A est complètement à revoir. De plus, ses coefficients non initialisés explicitement ne sont pas à zéro.
GERUSHTA	Valentyn	5	TB
GUESMIA	Zoheir	4.5	Il manque la transparence. Les coordonnées (0,0) sont bien dans l'image.
HARROUZ	Wail	4	L'usage de <code>overlay+padding</code> rend le code difficile à lire, bien qu'en fin de compte il fonctionne.
HUANG	Qijia	5	Le code ressemble beaucoup à celui de Weiqin Huang. OK pour cette fois, mais il me faudra une explication si ça se reproduit.
HUANG	Weiqin	4	Tu fais un “push” au lieu de “pull”. Il ne faut faire la moyenne des couleurs que sur la partie commune : ton test <code><20</code> dans <code>meanColor</code> n'est pas très satisfaisant.
LAZRAK	Mouad	0.5	Ce code n'a pas été compilé, et le principe du panorama ne semble pas avoir été compris.
LEMESLE-WELTI	Alban	4	Le test de superposition en testant la couleur noire est peu fiable et en plus est mal programmé. Je ne suis pas sûr que le devoir soit vraiment personnel, mais je laisse le bénéfice du doute pour cette fois.
MAHTAL	Ryad	0	PLAGIAT du devoir de W. Boumerdas
NDOKO	Arthur	5	TB
RAMOUL	Samy Rayan	2	La procédure de clic est très fastidieuse, avec 2 clics pour chaque point. L'ajout d'un <code>padding</code> dans l'image de panorama est inutile et néfaste. Après application de H, il faut diviser par la 3eme composante pour retrouver le point. Il faut s'assurer de ne jamais sortir d'une image.
SAULOU	Thomas	4	Les artéfacts sont dus à l'usage de la méthode <code>push</code> plutôt que <code>pull</code> .
SZYMCAK	Alan	4	Tu n'as pas bien compris comment marchait <code>anyGetMouse</code> , ce qui nuit à la flexibilité des clics. Il manque la transparence sur la zone commune.
ZHANG	Shihao	5	TB
Auditeurs libre			
MOLITOR	Pauline	4	Il manque la division par <code>X1[2]</code> (pour normaliser à 1) dans la fonction <code>panorama</code> . Ca explique que les résultats ne soient pas meilleurs. Code très clair.

Fundamental

Fundamenta(/5)

MVA

ALKAN	Gaël	5	TB. calculF et calculFmoindreC auraient pu être facilement fusionnées.
ANDREINI	Alexandre	4	Vue ta matrice A, tu es en fait en train de calculer la transposée de F. J'ai vérifié que le résultat est bien meilleur en prenant la transposée de tes calculs. Appliquer aveugélment la formule de recalcul de Niter peut provoquer une erreur si le nombre d'inliers tu 1er modèle est trop faible (pb numérique 1+epsilon=1).
AZIZIAN	Waïs	5	TB ! Difficile de dire si les épipoles sont faux parce qu'il sont dans l'image. En réalité, la position des épipoles est assez instable, bien que les lignes épipolaires soient assez précises.
BATARDIERE	Bastien	5	Je n'ai pas constaté de problème avec la minimisation aux moindres carrés, le résultat est bon. Quand tu écris upload, je pense que tu veux dire update (mettre à jour).
BENADY	Antoine		
BERGER	Clément	4.5	En fait, ton raffinement aux moindres carrés fonctionne bien modulo un truc très bête: l'opérateur [] ne peut pas prendre 2 arguments en C++, on doit utiliser (). Donc il faut des parenthèses pour remplir A et lire Vt.
COMBES	Theotime	5	TB !
DARCET	Thimothée	4.5	Bon ensemble, il y a simplement une confusion entre F et Ft dans l'affichage des lignes épipolaires. Pour les matrices de taille dynamique, utiliser Matrix et non Fmatrix.
DOS SANTOS ROCHA	Danilo		
ER-RAMMACH	Ilyes	4.5	Bien. Il y a potentiellement un pb avec le dénominateur du nb d'itérations, qui peut avoir un pb numérique si la proportion d'inliers est trop faible.
FERNANDEZ	Pierre	4.5	Bon travail, mais il faut être plus prudent dans l'application de la formule de mise à jour de Niter, pour éviter une division par zéro.
GALLAND	Lucie	5	TB. Il aurait été encore mieux de mettre dans une seule fonction le calcul avec 8 points et plus.
GRAIVE	Martin	4	Le problème numérique vient du fait que si le nombre d'inliers est trop faible au 1er tirage, on a $1-(m/n)^8 \approx 1$ et donc $\log=0$, d'où division par 0. Il manque le raffinement de F avec les inliers. Experimental::sample ne marche pas chez moi (gcc 7.5.0) et n'est certainement pas standard.
HAURET	Julien	2	Ecrire %mateches.size()-1 est une erreur car % est prioritaire sur -, comme une division. Du coup, on peut se retrouver avec l'indice -1, erreur mémoire. Il semble que ta matrice A est adaptée pour calculer la transposée de F. Le meilleur modèle n'est pas celui qui a la somme des erreurs la plus petite, mais le plus d'inliers. Il ne faut pas calculer AtA, la SVD de A suffit. La formule de Niter peut prendre une valeur -infini si on ne prend pas garde aux problèmes numériques. En tout cas, le résultat de l'affichage des lignes épipolaires montre de grosses erreurs.

Fundamental

JACOB	Paul	4.5	Il manque simplement le raffinement ultime de F avec tous les inliers.
LEGER	Victor	5	TB. Il faut noter qu'on a une classe Matrix qui fait de l'allocation dynamique et donc de taille inconnue lors de la compilation. Cela évite d'avoir à réserver 1000 lignes dans la matrice A.
LOTHE	Grégoire	3.5	Il manque l'optimisation de F aux moindres carrés avec les inliers. L'application de la mise à jour de Niter sans précaution peut conduire à une division par zéro numériquement. Il ne faut pas calculer $\text{transpose}(A) \cdot A$ mais calculer la SVD de A directement. Un clic dans l'image droite doit tracer une droite dans l'image gauche.
MAAZOUN	Imen	4.5	Il manque le calcul final aux moindres carrés. Appliquer aveuglément la formule pour Niter peut conduire à des problèmes numériques avec $\log(1-\epsilon)=0$.
MAISON	Lucas	5	TB. L'idée d'utiliser $\log 1p$ est bonne, elle évite une possible division par zéro.
MITTELMAN	Quentin	1.5	Le code est bien propre mais ne fonctionne pas à cause de 2 bugs: ligne 65, il faut <code>!alreadyPicked</code> et lignes 97-99 il faut <code>p1[1]</code> et non <code>p1[2]</code> . Il manque le forçage de F à être singulière par SVD, ainsi que la réestimation avec tous les inliers.
NGUYEN	Clément	5	TB. Il aurait été bien de factoriser le code d'estimation à partir d'un échantillon dans une fonction.
ORIOLE	Benoît	4.5	Bon travail, il manque simplement le raffinement aux moindres carrés de F avec tous les inliers.
REME	Raphael	5	TB !
RIOU	Auriane	4.5	C'est presque tout bon, sauf que tu mets un <code>srand(time(0))</code> à l'intérieur du while. C'est une erreur embêtante, car tant que tu restes dans la même seconde (résolution de la fonction time), tu tires le même échantillon. Tu testes ainsi au mieux 2 modèles. En sortant ce <code>srand</code> , tu aurais trouvé bien plus d'inliers.
SAUTIER	Corentin	5	TB. Factoriser le code pour le calcul de F à partir de 8+ points dans une fonction aurait été encore mieux.
SEGERIE	Charbel-Rapha	4.5	Il aurait fallu ajouter le raffinement de F aux moindres carrés avec tous les inliers. L'application aveugle de la formule pour Niter peut conduire à une division par zéro à cause d'une imprécision numérique.
SPINAT	Quentin	4	Il manque le raffinement avec tous les inliers. On peut avoir une division par 0 dans le recalcul du nombre d'itérations si le 1er tirage a trop peu d'inliers.
VANLAER	Youval	4	Je n'ai pas réussi à faire fonctionner <code>experimental::sample</code> . Bon travail dans l'ensemble, mais il manque l'optimisation aux moindres carrés. Le clic dans l'image droite ne fonctionne pas.

IMA

Fundamental

ACKER	Edgar	4	(devoir 2019) Tu as interverti F et sa transposee dans le trace des epipolaires. Forcer une valeur singuliere a 0 en fin de RANSAC est inutile puisque tu l'avais deja fait a chaque iteration. Ton distMax devrait etre son carre, puisqu'il est compare a la distance au carre. Le trace d'epipolaire suppose que les images ont meme largeur (c'est le cas pour les images donnees, mais ce n'est pas force).
ARGUELLO	Camillo	4	Je m'étonnais qu'il trouve plus de 900 inliers alors qu'il y a moins de 700 correspondances. En fait, tu oublies de vider inliers à chaque nouvelle estimation. De plus, l'affichage est trompeur car il donne inliers.size au lieu de bestInliers.size. Il manque le raffinement de F avec tous les inliers.
BOUDO	Louis	2.5	Je ne vois pas où tu normalises les points (par contre tu dénormalises bien la matrice). Il vaut mieux la SVD de A que de AtA, encore plus si les points ne sont pas normalisés. Diviser par norm(fx_) pour le calcul de la distance point-ligne est une erreur car la 3ème coordonnée ne doit pas être utilisée.
BOUMERDAS	Wassila	4	Ta matrice A est en fait celle qui calcule la transposée de F. Tu aurais trouvé bien plus d'inliers avec cette correction.
DI BON	Florian	1.5	Tu as une erreur bête ligne 84 : P1[1]=y2 au lieu de y1. Il faut calculer la SVD de A, c'est bien mieux numériquement que AtA. Tu oublies de denormaliser avec la matrice N. Il faut faire transpose(F)*x1 pour la ligne épipolaire à droite. Le calcul de Niter n'est pas compris. Le comportement étrange de ton programme est une erreur mémoire, les dimensions des vecteurs X1 et X2 n'ont pas été précisées lignes 184 et 189.
DJERADA	Nacer	4.5	C'est très bien, il faut juste être plus prudent dans le calcul de Niter qui peut devenir -infinity si m/n est trop petit au 1er tirage.
EL MADAFRI	Saad	4.5	La mise à jour sans précaution de Niter est risquée car on peut avoir un échantillon donnant très peu d'inliers et on a numériquement 1-eps=1 et on se retrouve avec Niter=-infinity.
FAURE	Maxime	4	Il vaut mieux la SVD de A que de AtA (meilleur conditionnement). Dans ton calcul de seuil, m/n sera toujours 0 car il fait la division euclidienne (ce sont des entiers). Ton seuil est censé faire baisser Niter, pas l'augmenter. Dommage qu'on ne puisse pas cliquer à droite pour voir la ligne épipolaire à gauche.
FORTIER	Victor	3.5	Ta matrice A calcule la transposée de F et non F. Pour le calcul de la distance point-ligne, tu oublies le dénominateur. Bien pour l'esprit critique dans le rapport. Mais attention, les centres optiques peuvent bouger peu et les épipoles être quand même dans l'image, c'est indépendant : seule la direction de la translation compte.

Fundamental

GAVALDA	Louis	2	Les résultats ne sont pas bons, et je n'ai pas trouvé toutes les sources d'erreur. Les coefficients de N hors diagonale ne sont pas mis à zéro. Vt est la transposée de V, donc c'est sa dernière ligne qui compte. Ta matrice A est adaptée en fait pour la transposée de F, mais ce n'est pas grave car tu corriges pour calculer les inliers. Pourtant, ce n'est pas le cas dans displayEpipolar.
GERUSHTA	Valentyn	3	Plusieurs bugs qui invalident les conclusions du rapport (mais c'est bien d'avoir investigué). La matrice A calcule la transposée de F. L'opérateur [] ne peut pas prendre deux paramètres (limitation C++), il faut utiliser les parenthèses. Pourquoi l'optimisation finale n'utilise-t-elle pas la SVD ?
GUESMIA	Zoheir	4.5	Bon travail. Il manque simplement l'optimisation finale de F avec tous les inliers.
HARROUZ	Wail	4	La matrice A que tu construis va être adaptée pour la transposée de F et non F. Il est inutile de raffiner aux moindres carrés à chaque fois qu'on améliore les inliers, le faire une fois toute à la fin suffit. Le recalcul de Niter peut donner -infinity à cause d'un problème numérique.
HUANG	Qijia	3.5	Il y a les mêmes défauts que pour Weiqin, mais je veux bien croire que les travaux sont personnels.
HUANG	Weiqin	3.5	La matrice A que tu écris calcule la transposée de F en fait. Tu aurais trouvé bien plus d'inliers sinon. Il manque l'optimisation avec tous les inliers à la fin. Il faut être plus prudent dans la mise à jour de Niter, on peut se retrouver avec -infinity si le premier échantillon a trop peu d'inliers. La fonction drawLine t'aurait épargné un peu de travail.
LAZRAK	Mouad	1	Ce code exprime une vague idée de la méthode, mais est loin de compiler.
LEMESLE-WELTI	Alban	2.5	Quand tu tires un échantillon, tu laisses simplement vide une case du tableau si l'élément choisi est déjà tiré, c'est une erreur, il faut insister. Ta matrice A est adaptée à la transposée de F. Le problème numérique dans Niter survient lorsque m/n est petit et élevé à la puissance 8, donc peut être ignoré face au 1. Il manque l'optimisation avec tous les inliers.
MAHTAL	Ryad	3.5	C'est dommage, tu renvoies le dernier F trouvé et non le meilleur. Ta formule de Niter est fautive, et en plus elle a un bug: m/size=0 car ce sont des entiers. Il faut m/double(size) par exemple,
NDOKO	Arthur	3.5	Quelques petits problèmes qui ont un impact négatif sur les résultats : tu multiplies les coordonnées par 0.01=1/100 et tu dénormalises par 1/1000; il est bien mieux de prendre la SVD de A que de AtA; ta matrice A calcule la transposée de F; diviser par la norme du vecteur représentant la ligne épipolaire est une erreur car c'est les 2 premières coordonnées dont il faut prendre la norme.

Fundamental

RAMOUL	Samy Rayan	2	Ca ne marche pas bien du tout à cause d'une erreur toute bête : ligne 176 on doit avoir $Df_prime(1,1)$. La mise à jour de niter est erronée. Avec ces corrections, ton code marche bien !
SAULOU	Thomas	2	Dans NormPoints, il manque l'initialisation de tmp à 0, il manque : la mise à jour du nb d'itérations et la réestimation de F avec tous les inliers. Un clic dans l'image droite n'affiche pas la ligne dans l'image gauche.
SZYMCZAK	Alan	4	Le code est mal indenté et difficile à suivre. Ta matrice A calcule la transposée de F.
ZHANG	Shihao	4.5	Bon travail. L'utilisation sans précaution de la formule de mise à jour des itérations est risquée car si le 1er échantillon donne très peu d'inliers, on a $1 - \epsilon = 1$ numériquement et une division par 0.

Auditeurs libre

MOLITOR	Pauline	1.5	Les mauvais résultats sont dus à plusieurs erreurs : la matrice A correspond à l'équation $x' F x = 0$ et non $x F x' = 0$; pour lire F, il faut la dernière ligne de $V = \text{transpose}(Vt)$; ligne 132, $X[1] = \text{matches.y1}$ et non $x2$; ligne 144, la division $\text{bestInliers.size()}/\text{matches.size()}$ se fait en type size_t, entier, donc le résultat est 0 (forcer la division en double avec un cast).
---------	---------	-----	--

Seeds

MVA		Seeds(/5)	
ALKAN	Gaël	4.5	Bon ensemble, mais il ne faut pas laisser les disparités se propager en dehors de l'intervalle.
ANDREINI	Alexandre	4	Tu ne vérifies pas assez rigoureusement l'inclusion des fenêtres dans l'image. L'inégalité stricte ne suffit pas, car on peut decaler de dMin.
AZIZIAN	Waïs	4.5	Il ne faut pas autoriser de disparités hors intervalle admissible lors de la propagation. Le y ne doit pas intervenir dans beginDisp et endDisp car on n'essaie de bouger qu'horizontalement.
BATARDIERE	Bastien	3	Tu ne prends aucune précaution pour t'assurer que les fenêtres ne débordent pas de l'image, c'est crucial quand on fait du traitement d'image. Il y a des bugs sérieux dans la propagation : best_incr=i au lieu de incr, cela doit être fait en-dehors de la boucle for(incr), et tu ne vérifies pas qu'on reste entre dMin et dMax. Le rapport rattrape en partie ces défauts.
BENADY	Antoine		
BERGER	Clément	5	TB !
COMBES	Theotime	3.5	La propagation est à revoir. Le test $d < win$ devrait être $x + d < win$. Du coup, souvent tu as dBest non initialisé et la propagation donne n'importe quoi.
DARCET	Thimothée	5	Pour avoir une carte vraiment dense avec nccSeed=-1, il aurait fallu initialiser bestNCC en conséquence et non à 0 dans find_seeds.
DOS SANTOS ROCHA	Danilo		
ER-RAMMACH	Ilyes	4.5	Bien, mais tu utilises l'hypothèse implicite que $dMax + win < 0$ pour ne pas déborder à droite, ce qui peut être faux. De plus, projeter sur l'intervalle [dMin,dMax] est moins bien que ne pas tester du tout en dehors de cet intervalle. Les fonctions min et max existent déjà dans la bibliothèque standard du C++, inutile de les redéfinir.
FERNANDEZ	Pierre	4.5	Ne pas autoriser des disparités hors intervalle [dMin,dMax] lors de la propagation. Sinon, tout est bien.
GALLAND	Lucie	5	TB. Dans find_seeds, initialiser score_max à -1 et non à 0 pour avoir une estimation dense dans le premier appel avec nccSeed=-1. Le code pour la sélection de la meilleure des 3 disparités lors de la propagation est fastidieux mais correct.
GRAIVE	Martin	5	Bien, mais l'hypothèse $d \leq 0$ n'est pas forcément toujours vérifiée, ça peut dépendre de la rectification ou d'un crop des images.
HAURET	Julien	4.5	Il ne faut pas laisser des disparités hors intervalle admissible se propager. A part ça, tout est bien.
JACOB	Paul	4.5	Code bien propre et rapport pertinent. Il est juste moins bien de prendre la valeur admissible de disparité la plus proche dans la propagation, au lieu de simplement ignorer les disparités en dehors.

Seeds

LEGER	Victor	4.5	Bien, mais tu utilises l'hypothèse implicite que $dMax+win < 0$ pour ne pas déborder à droite, ce qui peut être faux. De plus, projeter sur l'intervalle $[dMin, dMax]$ est moins bien que ne pas tester du tout en dehors de cet intervalle.
LOTHE	Grégoire	5	Dans <code>find_seeds</code> , il est trop conservateur de ne pas considérer les points dont une des disparités donnerait une fenêtre qui déborde, mais c'est très bien à part ça.
MAAZOUN	Imen	5	TB !
MAISON	Lucas	4	C'est bien d'avoir voulu accélérer. Mais en fin de compte, c'est des efforts pour pas grand chose : il ne faut pas chercher en dehors de $[dMin, dMax]$ dans <code>find_seeds</code> , ça aurait été bien plus efficace. En plus, tu as un bug dans tes <code>summed area tables</code> , j'ai vérifié en remplaçant la fonction <code>sum</code> par celle initiale.
MITTELMAN	Quentin	4.5	Un petit bug : ligne 102, $y < win$ au lieu de \leq . Le choix de ne pas calculer de disparité pour les points dont l'une des disparités possibles donnerait une fenêtre incomplète est trop conservateur dans <code>find_seeds</code> .
NGUYEN	Clément	5	TB !
ORIOLE	Benoît	4.5	Il est préférable de directement ne pas tester les disparités en-dehors de l'intervalle admissible plutôt que projeter après. Dans la propagation, on a potentiellement aucune entrée dans la boucle <code>for</code> et alors <code>bestd</code> n'est pas initialisé, le <code>if(bestd < dMin)</code> qui suit est indéterminé. Enfin, tu utilises l'hypothèse implicite que $dMax+win < 0$, ce qui n'est pas forcément le cas.
REME	Raphael	5	TB !
RIOU	Auriane	5	L'utilisation du tableau <code>d_neighbourhood</code> est judicieuse, mais son <code>sizeof</code> est en fait 12: taille en octets de 3 entiers. Pour le nombre d'éléments, il faut diviser par <code>sizeof(int)</code> . Ne pas laisser se propager des disparités hors intervalle admissible. Ces défauts sont compensés par un très bon rapport !
SAUTIER	Corentin	5	TB ! Il vaut mieux ne pas tester du tout les disparités hors intervalle plutôt que de prendre la plus proche lors de la propagation.
SEGERIE	Charbel-Rapha	4.5	Ta condition d'arrêt dans la boucle <code>for(d)</code> de <code>find_seeds</code> est trop conservatrice, elle supprime une marge trop grande à la gauche de l'image. Lors de la propagation, il est mieux de ne jamais tester en dehors des bornes <code>dMin</code> et <code>dMax</code> plutôt que de faire un <code>clip</code> après.
SPINAT	Quentin	3.5	Tes fonctions <code>sum</code> et <code>correl</code> bouclent jusqu'à <code>win</code> exclus , ce qui a le défaut mineur de faire une fenêtre non centrée, mais plus gênant est le fait que la division se fait par $(2*win+1)^2$ pour la moyenne, ce qui ne correspond pas à la taille de ta fenêtre. Initialiser <code>bestNCC</code> à -1 pour avoir vraiment une estimation dense. La propagation ne prend pas garde à <code>dMax</code> et pas assez rigoureuse pour ne pas risquer de sortir de l'image.

Seeds

VANLAER	Youval	4.5	L'ensemble est bon mais comporte quelques petits défauts : nccSeed n'est pas utilisé dans find_seeds ; dans la propagation, rien ne garantit que $x+s.d+1+win$ ne va pas déborder de l'image ; il vaut mieux vérifier les disparités admissibles dès le début que projeter dans l'intervalle après coup dans la propagation. De plus, l'hypothèse $dMax \leq 0$ est ici vraie mais pourrait ne pas l'être.
IMA			
ACKER	Edgar	2	Il y a de multiples bugs qui provoquent de mauvais résultats. Dans correl, les indices pi et pj doivent aller jusqu'à win inclus. De plus, tu t'es trompé entre indices pi et pj pour l'accès aux pixels. Dans propagate, le -1 que tu ajoutes à d+shift est une erreur. De plus, il faut aller jusqu'à d=1 inclus.
ARGUELLO	Camillo	2.5	Tu ne t'es pas rendu compte que nccSeed n'était pas utilisé par find_seeds ? En plus, tu n'insères pas dans la file de priorité les graines, donc pas de propagation possible.
BOUDO	Louis	3.5	Dans correl, les indices doivent aller jusqu'à win inclus. Dans find_seeds et propagate, tu as une bande à gauche de l'image fausse car si le if dans la boucle for est toujours faux (fenêtre qui débord), tu mets quand même une disparité.
BOUMERDAS	Wassila	4.5	Ton find_seeds ne s'approche pas assez du bord gauche, car si $x+dMin$ donne une fenêtre qui sort, tu ne testes pas les autres disparités. Dans la propagation, il est moins bien de forcer après coup à rester dans $[dMin, dMax]$ au lieu de simplement ignorer les disparités hors intervalle. Bravo pour ta persévérance à trouver ton bug.
DI BON	Florian	4.5	Lors de la propagation, tu ne vérifies pas que la fenêtre en $x+s.d+dif_disp$ est bien entièrement dans l'image 2. Ton "inversion" 3D est due à une ligne de show3D qui a disparu...
DJERADA	Nacer	5	Tu as un tout petit problème : tu testes $s.d+2$ lors de la propagation au lieu de t'arrêter à $s.d+1$. Si tu mettais un $<$ au lieu de \leq dans ta boucle for, ce serait OK.
EL MADAFRI	Saad	5	TB !
FAURE	Maxime	2.5	Lorsque tu es trop près du bord gauche de l'image, currentCorrel n'est pas initialisé mais comparé à max. Il manque un 'continue' qui ferait passer à la disparité suivante si la condition $X < win$ n'est pas vérifiée. Il manque la propagation.
FORTIER	Victor	4.5	Les trous qui restent après propagation sont dus à une valeur hors des bornes $[dMin, dMax]$. Il faut faire attention de ne pas déborder. C'est bien d'avoir testé avec une autre paire, mais les bornes dMin et dMax ne sont pas adaptées pour cette paire !
GAVALDA	Louis	3.5	Tu as un bug idiot qui m'a pris beaucoup de temps à trouver : dans correl, tu fais $s1+=$ et $s2+=$ au lieu de $s1=$ et $s2=$. Tu as un deuxième bug (moins grave) dans propagate, avec comme condition dans le for $k < +1$ au lieu de $k \leq +1$.

Seeds

GERUSHTA	Valentyn	2	Tu ne t'es as aperçu que les résultats que tu obtenaient étaient absurdes ? C'est que ta formule de la corrélation est complètement à revoir (numérateur faux et bornes s'arrêtant à win-1). Le reste est mieux. Attention de ne pas dépasser des bornes de disparité lors de la propagation.
GUESMIA	Zoheir	3.5	Tu as un bug dans le calcul du dénominateur de la NCC : ce que tu calcules n'a aucun sens, tu prends des racines carrées de valeurs que tu viens d'élever au carré (soit la valeur absolue) et tu multiplies entre images 1 et 2. Les résultats auraient été meilleurs sans ce bug.
HARROUZ	Wail	3.5	C'est dommage, le code était presque tout bon. Simplement, dans la propagation, ton test de "fenêtre dans image" est inexact car il manque x. Tu aurais pu t'apercevoir que la propagation donnait de très mauvais résultats et investiguer.
HUANG	Qijia	4	Dans la propagation, tu as des pixels sur le bord gauche qui ne vont pas sauter le if dans la boucle for et qui utiliseront donc une valeur bestd non initialisée. D'autre part, il est préférable de ne pas tester du tout les disparités hors bornes plutôt que projeter après coup. L'hypothèse que les fenêtres ne débordent pas vers la droite est vraie ici, mais ça pourrait ne pas être le cas.
HUANG	Weiqin	3	Je ne comprends pas bien pourquoi tu prends des valeurs absolues aussi souvent, surtout pour les disparités. Du coup, il y a un bug dans correl car dans le produit $a*b$, il se peut que les signes (sans les valeurs absolues) soient différents. As-tu vérifié que l'utilisation d'images cache pour les sommes accélèrent l'algorithme ? Je doute.
LAZRAK	Mouad	0.5	Tu n'as clairement pas essayé de compiler ton programme. Coder sans tester est un exercice quasi impossible. Ce n'est pas la bonne façon de faire, il faut programmer un peu, tester, etc, et non pas essayer de tout faire d'un coup.
LEMESLE-WELTI	Alban	3.5	Il y a des bugs. Dans correl, tu n'initialises pas le vecteur S à 0. Dans find_seeds, tu ne testes pas les disparités suivantes si à $x+dMin$ tu n'as pas une fenêtre incluse dans l'image (il faudrait continue plutôt que break). Lors de la propagation, tu ne vérifies pas que tu restes dans les bornes $[dMin, dMax]$.
MAHTAL	Ryad	5	Bien. Dans find_seeds et propagate, ta vérification de non-débordement à droite n'est pas assez stricte, ça devrait être $\geq im2.width$ et non $>$. Dans propagate, tu as quelques pixels à gauche de l'image qui ont une disparité arbitraire car d n'est pas initialisé et si la boucle for n'est valide pour aucune des 3 disparités testées, tu utilises cette valeur arbitraire.

Seeds

NDOKO	Arthur	3.5	Dans les fonctions correl et sum, il faut aller jusqu'à win inclus, c'est important car sinon ça fausse le calcul de la moyenne. Lors de la propagation, attention de ne pas déborder de l'intervalle [dMin,dMax]. L'aspect quantifié des profondeurs est dû au fait que les disparités calculées le sont aussi, au pixel près.
RAMOUL	Samy Rayan	4	Tu as des problèmes dans les indices supérieurs de boucles for. Dans correl, il faut aller jusqu'à win inclus. Dans find_seeds, il ne faut pas dépasser dMax. Dans propagate, ne pas aller jusqu'à fin+1. Lors de find_seeds, même si la fenêtre en x+dMin débord, certaines suivantes peuvent ne pas le faire, il manque donc une bande d'environ 20 pixels à gauche.
SAULOU	Thomas	4	Quelques petits défauts dans le programme : la fonction sum considère une fenêtre 7x7, alors qu'on a pris 9x9 ; initialiser bestNCC à -1 pour avoir une carte vraiment dense ; lors de la propagation, ne pas sortir des bornes [dMin,dMax].
SZYMCZAK	Alan	3.5	Ton crash est dû à une fenêtre qui sort partiellement de l'image, tes vérifications ne sont pas exactes. En mode debug, il y a une assertion (instruction assert) qui ne passe pas. Avec le debugger, ça permet de voir où le problème se produit. Il faut aussi vérifier que tu ne propages pas de disparité hors de l'intervalle [dMin,dMax].
ZHANG	Shihao	Plagiat	C'est une tentative de masquer un corrigé d'une année précédente, je refuse de noter ce devoir.
Auditeurs libre			
COULIN	Jade		Bon devoir. Il faut quand même vérifier dans la propagation de ne pas sortir des bornes [dMin,dMax]. Attention de ne pas faire 2 fois le même calcul (utiliser une variable) : if(... ccorrel(..)) ...=ccorrel(...). Ligne 171 le test devrait être <im2.width(). Je crois que tu es en thèse et n'as pas besoin de note ?
MOLITOR	Pauline	4.5	C'est parfait, si ce n'est qu'il faut empêcher la disparité de sortir des bornes lors de la propagation.

GCDisparity

GCDisparity(/5)

MVA

ALKAN	Gaël	4.5	Il faut s'assurer que l'appel à zncc ne fait pas sortir de l'image, ce qui arrive pour certaines disparités près du bord droit. Inutile de se fatiguer pour K, toute valeur suffisamment grande convient et donne le même résultat.
ANDREINI	Alexandre	4.5	Appeler zncc sans vérifier qu'on reste dans l'image est une erreur. Il y a quelques points noirs sur le bout du nez, ce qui suggère qu'il y a un autre bug, certainement parce que c'est les points de plus forte disparité. J'ai vérifié en augmentant dmax un peu, ça règle le problème.
AZIZIAN	Waïs	5	Bravo !
BATARDIERE	Bastien	4	Pour les pixels près du bord droit, certaines disparités font sortir de l'image, attention de ne pas appeler zncc. Curieusement, tu mets (lambda,0) avec le voisin de droite, alors que tu mets bien (lambda,lambda) avec celui du bas.
BENADY	Antoine		
BERGER	Clément	5	TB ! Inutile de faire de la subtilité avec K, le prendre suffisamment grand garantit un résultat constant. Il est plus efficace de faire un poids de régularité (lambda,lambda) avec un voisin vertical et un horizontal plutôt que (lambda,0) avec les 4 voisins.
COMBES	Theotime	4	Multiplier d par zoom est une erreur, il ne faut pas de parenthèse dans la multiplication. Attention aussi de ne jamais appeler zncc quand ça fait sortir de l'image.
DARCET	Thimothée	5	Rien à redire, c'est très bien.
DOS SANTOS ROCHA	Danilo		
ER-RAMMACH	Ilyes	5	TB. Il n'est pas nécessaire de recalculer un K pour chaque noeud, dès qu'il est suffisamment grand le résultat ne doit plus changer.
FERNANDEZ	Pierre	4.5	Attention de ne pas sortir de l'image avec certaines disparités près du bord droit.
GALLAND	Lucie	4.5	Certaines disparités peuvent faire sortir de l'image, il ne faut pas appeler zncc dans ce cas, mais mettre un poids infini pour empêcher la coupure.
GRAIVE	Martin	3.5	La multiplication de d par zoom est un bug, qui a pour effet d'aplatir le visage (disparité correcte divisée par zoom=2). Vérifier qu'on ne sort pas de l'image avant d'appeler zncc.
HAURET	Julien	4.5	Il faut prendre garde de ne pas appeler zncc lorsque cela fait sortir de l'image (il n'y a pas de garde-fou). Mettre un poids infini le cas échéant règle le problème.
JACOB	Paul	4.5	Ne pas appeler zncc quand cela fait sortir de l'image 2. Mettre un poids infini est une bonne solution. Pas besoin de se fatiguer avec K, dès qu'il est suffisamment grand, le problème est réglé.
LEGER	Victor	3.5	Faire $\text{zoom}^*(x+d)$ est une erreur : on ne teste que les disparités de 2 en 2. En plus, il faut vérifier qu'on ne sort pas des bornes de l'image avant d'appeler zncc.

GCDisparity

LOTHE	Grégoire	4.5	La fonction zncc ne doit pas être appelée si ça fait sortir de l'image de droite. Mettre un poids infini dans ce cas est la bonne solution.
MAAZOUN	Imen	4.5	Il faut faire attention de ne pas sortir de l'image avec certaines disparités. Inutile de recalculer K pour chaque noeud, le résultat ne varie plus quand il est suffisamment grand.
MAISON	Lucas	4	Près du bord droit de l'image, des disparités font sortir de l'image. Il ne faut pas appeler zncc dans ce cas, mais mettre un poids infini pour empêcher la coupure à cet endroit. Le fait de mettre les coûts de régularité (λ , λ) dans les deux sens entre voisins revient à doubler λ .
MITTELMAN	Quentin	3.5	Il faut vérifier que certaines disparités ne font pas sortir de l'image. Si tu fais $\text{zoom} \times (x + d_{\max})$, d_{\max} est aussi multiplié par zoom, ce qui n'est pas souhaité.
NGUYEN	Clément	3.5	Il ne faut pas multiplier d par zoom ! De plus, il faut vérifier avant d'appeler zncc qu'on ne sort pas de l'image. Dans le cas contraire, il faut mettre un poids infini pour empêcher la sélection d'une disparité hors bornes.
ORIOLE	Benoît	4.5	Il ne faut pas appeler zncc lorsque la fenêtre sort de l'image, et ça arrive pour certaines disparités et certains pixels près du bord droit.
REME	Raphael	5	Parfait !
RIOU	Auriane	4.5	Il faut simplement faire attention de ne pas appeler zncc pour les disparités qui font sortir de l'image. Mettre un poids infini à la place règle le problème:
SAUTIER	Corentin	4	Il faut être beaucoup plus prudent de ne pas sortir de l'image avec certaines disparités. Il est plus efficace de mettre les deux capacités (λ , λ) entre (i, j, d) et $(i+1, j, d)$ plutôt que (λ , 0) dans chaque sens.
SEGERIE	Charbel-Rapha	5	Tu as bien fait de vérifier qu'on ne sort pas de l'image avec zncc, mais il faut mettre un poids infini pour empêcher sa sélection, non wcc.
SPINAT	Quentin	5	Il faut vérifier avant d'appeler zncc qu'on ne sort pas de l'image. Mais tu as un bonus pour avoir essayé sur tes propres données, ça compense. Bravo pour ton effort !
VANLAER	Youval	4.5	Il faut stopper la chaîne des disparités k avant nd pour les pixels près du bord de l'image, pour ne pas calculer un zncc qui sort de l'image.
IMA			
ACKER	Edgar	0	non rendu
ARGUELLO	Camillo	4.5	Il faut se montrer prudent et ne pas appeler zncc quand la disparité fait sortir de l'image. C'est bien d'avoir testé plusieurs valeurs de λ .
BOUDO	Louis	4.5	Les appels à zncc ne doivent pas être systématiques, certaines disparités peuvent faire sortir de l'image.
BOUMERDAS	Wassila	4	Attention, les appels à zncc peuvent faire sortir de l'image, il faut vérifier avant. Tu n'as pas vraiment suivi les consignes avec la fonction rho.

GCDisparity

DI BON	Florian	4	Ne pas appeler zncc systématiquement, certaines disparités peuvent faire sortir de l'image. Tu aurais dû utiliser un Graph avec des int, c'est plus rapide.
DJERADA	Nacer	3.5	Ne pas multiplier d par zoom, ce n'est pas correct. Attention de ne pas appeler zncc quand la disparité fait sortir de l'image. Il faut revoir tes t-links, l'ordre des poids est inversé et mettre INF au lieu de 0 empêche de sélectionner les disparités extrêmes.
EL MADAFRI	Saad	4.5	Certains appels à zncc font sortir des bornes de l'image, il faut y faire attention.
FAURE	Maxime	3.5	Les add_tweights ne doivent pas être appelés plusieurs fois sur le même noeud, il faut donc sortir de la boucle for(couche). Les appels à zncc doivent être vérifiés avant pour ne pas sortir de l'image.
FORTIER	Victor	4.5	Il ne faut pas multiplier la disparité d par zoom. Ne pas calculer zncc quand la disparité fait sortir de l'image. Bonus pour un très bon rapport !
GAVALDA	Louis	4	Tu inverses l'ordre des poids dans les t-link. De plus, tu dois mettre des 0 et non INF. Attention, ne pas appeler zncc quand ça fait sortir de l'image.
GERUSHTA	Valentyn	4.5	Certains appels à zncc ne devraient pas avoir lieu car ils font sortir de l'image. Il faut remplacer par un poids infini.
GUESMIA	Zoheir	4	Dans certains appels, tu multiplies d par zoom (mais pas toujours). C'est une erreur. Il ne faut pas appeler zncc quand la disparité fait sortir de l'image.
HARROUZ	Wail	4.5	Des appels à zncc font sortir de l'image, il faut vérifier la disparité avant.
HUANG	Qijia	0	non rendu
HUANG	Weiqin	5	TB !
LAZRAK	Mouad	2	Il y a pas mal de choses à revoir dans l'attribution des poids du graphe. Tu n'as pas réservé assez de noeuds, ça fait un crash du programme chez moi (et une assertion qui ne passe pas en debug). Enfin, attention aux appels à zncc, certains peuvent sortir de l'image.
LEMESLE-WELTI	Alban	4	La disparité ne doit pas être multipliée par zoom. D'autre part, il ne faut pas appeler zncc quand cela fait sortir de l'image.
MAHTAL	Ryad	4	Il est assez maladroite d'utiliser des constantes comme 44 et 45 dans le code, il faut utiliser dmin et dmax. Attention de ne pas appeler zncc quand la disparité fait sortir de l'image.
NDOKO	Arthur	4.5	Prudence, des appels à zncc font sortir de l'image. Il faut d'abord vérifier.
RAMOUL	Samy Rayan	4.5	Les appels à la fonction zncc doivent être filtrés pour ne pas sortir de l'image, ce qui arrive près du bord droit avec des disparités trop grandes.
SAULOU	Thomas	4	Les appels à zncc doivent être filtrés pour ne pas risquer de sortir de l'image (il est judicieux de mettre des poids infinis dans ce cas). Mettre (lambda,lambda) sur les 4 voisins revient à avoir des poids 2*lambda entre voisins.

GCDisparity

SZYMCZAK	Alan	4	Il ne faut pas multiplier d par zoom, ça n'a pas de sens. Ne pas appeler zncc quand cela fait sortir de l'image.
ZHANG	Shihao	4.5	Les cas $i < n_x - 1$ et $j < n_y - 1$ doivent être traités séparément. Un warning car on n'utilise pas un <code>Graph<double,double,double></code> mais un <code>Graph<int,int,int></code>
Auditeurs libre			
COULIN	Jade		Tout est bon, il faut juste faire attention de ne pas appeler zncc lorsque cela fait sortir de l'image. Mettre un poids infini dans ce cas règle le problème.
MOLITOR	Pauline	2	Tu n'as pas compris la construction du graphe: on ne calcule jamais explicitement $ dp-dq $, les différentes couchent dans le graphe codent cela. Ta fonction <code>estimate_d</code> est inutile et nuisible. Attention aussi de ne pas appeler zncc quand la disparité fait sortir de l'image.