

3DCV 2024

MVA		Grade (/20)	TP1	TP2	TP3	TP4	Exam
BELLANCA	Ugo	14.5	5	5	3.5	4.5	9
FOURREAU	Félix	12	4	4	4.5	3	7
KADDAMI	Yassine	13.5	5	5	5	4	5
MEGHRAOUI	Abdallah	14.5	5	5	5	4.5	7.5
MICHETTI	Marius	13	5	2	5	4	8.5
RAVAUD	Tom	15.5	5	4	5	5	10.5
SAULNIER	François		4.5				
SEREYJOL-GARROS	Nicolas	13.5	5	4.5	4.5	4	6.5
SOUCAILLE	Matti	12.5	5	4.5	4.5	3.5	4.5
VERSINI	Lucas	18.5	5	5	5	5	16
VICTOR	Ludvig	15.5	4	4.5	5	5	11.5
WAUQUIEZ	Mathis	13	4	4	3	5	9

IMA		Grade (/20)	TP1	TP2	TP3	TP4
CADI	Yacine	14.5	5	4.5	4	1
DRACEA	Iulian-Ilie	15.5	4	2	4.5	5
DROZDOV	Dmitrii	17	5	3.5	4	4.5
GONCALVES	Samuel	17	5	3	4.5	4.5
GRUSS	Carlos	17	5	4	4	4
HASSAYOUNE	Ahmed	18	4.5	4.5	5	4
KHAREF	Okba	14	5	0	4.5	4.5
KILINC	Sena	16.5	4.5	4	4	4
LAVAL	Luka	17.5	5	4.5	4	4
MAOUCHE	Mounir	13	4	2	3	4
OUMAZIZ	Thiziri	13.5	5	4.5	4	0
PAGNEUX	Gabriel	13	4.5	4	3	1.5
RCHAKI	Oussama	17.5	5	4.5	4	4
ROUSSELLE	Naomi	18	5	4	4.5	4.5
SELLAHENNEDI	Ménalie	12	4	1.5	1.5	5
SHIKHLI	Nadir	18.5	5	4	5	4.5
SIDKI	Noureddine	16	4	3.5	4	4.5
SOARES	Robin	17	4.5	4	4.5	4
TIAN	Ning	17.5	5	4	4.5	4

Panorama

Panorama(/5)

MVA

BELLANCA	Ugo	5	Good work, even if there may be white pixels in image 1 and detecting superposition through check of color may make mistakes.
FOURREAU	Félix	4	Good for trying progressive blending. The grid you see is because you “push” pixels from I1, instead of “pulling” from them. The grid-filling part seems to have no effect...
KADDAMI	Yassine	5	Good!
MEGHRAOUI	Abdallah	5	Good! Notice though that coordinates (0,0) are still valid for a pixel, so your function isWithinBounds is a bit conservative.
MICHETTI	Marius	5	Good, though detecting superposition through white color is sensitive to white color in image 1.
RAVAUD	Tom	5	Very good!
SAULNIER	François	4.5	Good, it lacks just the transparency blending on common parts of images.
SEREYJOL-GARROS	Nicolas	5	Great work and very good report!
SOUCAILLE	Matti	5	Good overall, though the code is a bit more complex than necessary. Notice that coordinates (0,0) are still a valid pixel.
VERSINI	Lucas	5	Excellent overall, both code and report.
VICTOR	Ludvig	4	Color distortion is due to overflow of unsigned char in Color+Color: it wraps around to 0 when above 255. The point selection procedure is not very user friendly.
WAUQUIEZ	Mathis	4	Transparency is lacking (your trial did not work because of overflow of sum of unsigned char). You assume the (0,0) pixel of left image coincides with (0,0) pixel of panorama, which is not necessarily true.

IMA

CADI	Yacine	5	Neat and clean, good job!
DRACEA	Iulian-Ilie	4	The white grid you see in your panorama is because your push pixels instead of pulling them. Using white color to detect superposition is not optimal.
DROZDOV	Dmitrii	5	Good!
GONCALVES	Samuel	5	Good!
GRUSS	Carlos	5	Very good and great to see results on your own pair of images.
HASSAYOUNE	Ahmed	4.5	The grid appearing is due to pushing pixels instead of pulling. Relying on background color for superposition fails sometimes (though rarely).
KHAREF	Okba	5	Excellent! Nice to see the results on other test images.
KILINC	Sena	4.5	Calculating inverse(H) for each pixel takes its (small) cost. Right click is detected through event.button.
LAVAL	Luka	5	Good! You could check the button clicked through Event::button.

Panorama

MAOUCHE	Mounir	4	The grid that appears in the panorama is because you push pixels from I1 instead of pulling. Using white color to detect superposition is not the best thing to do.
OUMAZIZ	Thiziri	5	Good job, but relying on a particular background color for detecting superposition is not optimal as it is a legit color in image 2.
PAGNEUX	Gabriel	4.5	Good, though relying on color check for superposition is not the best option, as the information is already present. It would have been good if more than 4 pairs could be input.
RCHAKI	Oussama	5	Very good! I would have liked to see the results of the experiment of your desk images, as promised in the report...
ROUSSELLE	Naomi	5	Good job!
SELLAHENNEDI	Ménalie	4	The grid you see in the panorama is because you push pixels from image 1 instead of pulling. Detecting superposition based on background color is not foolproof.
SHIKHLI	Nadir	5	Good job!
SIDKI	Noureddine	4	Your code relies on the fact that point (0,0) of image 1 and the panorama coincide. It may be true for the test images, but not in general. Good otherwise.
SOARES	Robin	4.5	Detecting superposition through color check is a pity, as the information is already present through earlier check.
TIAN	Ning	5	Great!

Fundamental

Fundamental(/5)

MVA

BELLANCA	Ugo	5	Very good!
FOURREAU	Félix	4	Good, though there is a bug that degrades the performance: when you count inliers, you compute $F \cdot \text{points1}[i]$ instead of $\text{transpose}(F) \cdot \text{points1}[i]$.
KADDAMI	Yassine	5	Very good!
MEGHRAOUI	Abdallah	5	Very good!
MICHETTI	Marius	2	Several mistakes that ruin the results: 1) method <code>setDiagonal</code> just sets the diagonal but does not ensure there are 0 outside. 2) Dynamic setting of <code>Niter</code> must ensure that <code>m</code> is large enough so that $1 - (m/n)^8 < 1$ numerically. Computing <code>m/matches.size()</code> is Euclidean division, yielding 0. Operands must be cast into floating point.
RAVAUD	Tom	4	There is a dumb bug (but aren't they all?) that distorts the results: $A(i,4) = x[1] \cdot y[1] \cdot y[1]$. Fixing that yields much better results.
SAULNIER	François		
SEREYJOL-GARROS	Nicolas	4.5	Good, but numerical precautions must be taken when applying formula for update of <code>Niter</code> .
SOUCAILLE	Matti	4.5	The only thing lacking is the least square refinement of <code>F</code> based on all inliers at the end.
VERSINI	Lucas	5	Very good! Note though that your trial on the first homework images is not very pertinent, because there is no parallax and $F=0$...
VICTOR	Ludvig	4.5	Very good, except for a dumb bug in <code>getDistance</code> : line 94 should be <code>x[1]=match.y1</code> , not <code>y2</code> ! Fixing this increases significantly the number of inliers and yields a better <code>F</code> .
WAUQUIEZ	Mathis	4	Formula for adjustment of <code>Niter</code> must not be applied without numerical precaution, as it may stop the loop just after first iteration if <code>inliers.size() << matches.size()</code> . At the end of RANSAC, there should be a refinement based on all inliers.

IMA

CADI	Yacine	4.5	There should be refinement of <code>F</code> based on all inliers at the end of RANSAC. <code>1+rand()@n_match</code> is dangerous as the result is in <code>[1,n_match]</code> whereas index should be between 0 and <code>n_match-1</code> .
DRACEA	Iulian-Ilie	2	At output of SVD, we use the last column of <code>V</code> , which is the last row of its transpose, that we get as output of function <code>svd</code> . When adjusting the number of iterations, <code>Niter_lower_bound</code> may overflow the maximal integer, yielding a negative number <code>Niter</code> . Since your index <code>i</code> is <code>size_t</code> , the algorithm may run indefinitely. Moreover, in this formula, the division must be in floating point.
DROZDOV	Dmitrii	3.5	When finding inliers, you multiply left image points by <code>F</code> instead of its transpose, which yields wrong results. The numerical precautions you take for updating <code>Niter</code> are not sufficient, as the result can overflow the integers.

Fundamental

GONCALVES	Samuel	3	Based on the construction of matrix A, point in left image must be multiplied by transpose(F) (as in the course). At iteration 0, you may have very few inliers and Niter becoming -infinity because of numerical problem. There should be a refinement based on all inliers at the end.
GRUSS	Carlos	4	Your formula for displaying the epipolar line in right image (clicking in left image) is wrong. Rather than setting a fixed number of inliers as precaution before adjusting Niter, it should rather be a proportion.
HASSAYOUNE	Ahmed	4.5	Works fine, but there is a potential pitfall: if A is 8x9, the compact SVD (default for function svd) gives Vt of size 8x9, so that the last row of Vt is not computed...
KHAREF	Okba	0	This is not your personal work, it was copied from an earlier year.
KILINC	Sena	4	The update of Niter should have tighter check of numerical errors, I had many runs finishing at iteration 0. There should be refinement with all inliers at the end for F. Multiplying an uninitialized matrix by 0 may still leave some NaN.
LAVAL	Luka	4.5	At the end of RANSAC, a refinement based on inliers should be performed. You should not rely on multiplying an uninitialized matrix by 0 to set the null matrix: one of the coefficients could be Nan and stay NaN after multiplication. Use method fill.
MAOUCHE	Mounir	2	If you use a sample-specific normalization, you must store the normalization parameters so as to de-normalize after. EstimateF should consider more than the 9 first matches when useRefinement is used.
OUMAZIZ	Thiziri	4.5	Good, but if A is 8x9, function svd uses by default the compact mode so that Vt is 8x9 and column number 8 of V is not available.
PAGNEUX	Gabriel	4	The last line of A is left uninitialized, which may cause trouble. There is no refinement with all inliers at the end of RANSAC.
RCHAKI	Oussama	4.5	Your function getOptimalIterations always returns 100000, because your inlier ratio formula is actually the inverse of its true value...
ROUSSELLE	Naomi	4	Good, except that in displayEpipolar you invert the roles of F and its transpose, hence the epipolar lines are slightly wrong. There should be a refinement with all inliers at the end.
SELLAHENNEDI	Ménalie	1.5	You should have suspected something was wrong since you find no outlier. If you don't use a fixed normalization, you must store the normalization matrix so as to de-normalize correctly. Your formula for Niter is wrong. The way you build A, points in left image must be multiplied by F, not its transpose (as in your report).
SHIKHLI	Nadir	4	You invert the role of F and its transpose in displayEpipolar. Moreover, when clicking on right image, x of click should be relative to I1.width(). Your precaution against numerical error of Niter is wrong.
SIDKI	Noureddine	3.5	You should initialize N with N.fill(0), otherwise you may have random coefficients left. There is no adjustment of Niter. At the end of RANSAC, there should be a refinement based on all inliers.

Fundamental

SOARES	Robin	4	You should take the SVD of A , not of $\text{transpose}(A)*A$, because the former is better conditioned. It seems your distance threshold for inlier/outlier is $\text{sqrt}(\text{distMax})$.
TIAN	Ning	4	In the update of Niter, numerical problems may occur. At the end of RANSAC, refinement with all inliers should be performed. When clicking on right image, $l1.\text{width}()$ must be subtracted from x of click.

Seeds

Seeds(/5)

MVA

BELLANCA	Ugo	3.5	No propagation since you compare again to nccSeed before committing a new disparity. Using ccorrel would have spared you some code and for norm of patch you could have used ccorrel with same image.
FOURREAU	Félix	4.5	During propagation, it is not as good to simply project into [dmin,dmax] as to just ignore disparities outside the interval. Also, beware of overflow in x (could happen if dmax were positive).
KADDAMI	Yassine	5	Good!
MEGHRAOUI	Abdallah	5	Good!
MICHETTI	Marius	5	Very good, neat and tidy!
RAVAUD	Tom	5	Good job!
SAULNIER	François		
SEREYJOL-GARROS	Nicolas	4.5	Your function find_seeds is a bit too conservative as it discards pixels on the left for which some disparities (but not all) lead to a suitable patch.
SOUCAILLE	Matti	4.5	Valid disparities are assigned to some pixels for which no suitable patch is available. The safety checks in correl and sum overload the code without benefit, as calls are filtered before.
VERSINI	Lucas	5	Excellent! Very good code and amazing report!
VICTOR	Ludvig	5	Good code+report, though you should protect from division by zero (uniform patch) in NCC.
WAUQUIEZ	Mathis	3	In find_seeds, it makes no sense to compute NCC for disparity 0. In propagate, bestNcc should be initialized to -1 or lower, not 0. Also, you do not check if disparity stays within [dmin,dmax].

IMA

CADI	Yacine	4	In find_seeds, pixels yielding no suitable corresponding patch for any disparity get assigned uninitialized bestDisparity (only for dense case). In propagate, check also overflow on the right (could happen if dmax>=0).
DRACEA	Iulian-Ilie	4.5	Pixels with no valid corresponding patch still get assigned dmin as disparity, both during dense computation and propagation.
DROZDOV	Dmitrii	4	In function sum, dx=win and dy=win are discarded, which is a mistake and makes NCC not bounded by [-1,1]. Fixing that selects much fewer seeds, as it should. Patch inclusion must be checked through image2.width, not image1.
GONCALVES	Samuel	4.5	The only defect is that during propagation some pixels with no valid translated patch still get assigned a disparity.

Seeds

GRUSS	Carlos	4	In the dense case, pixels with no valid patch still get disparity dmin. Why does the relative shift in propagation reach +2? Why the threshold -0.9 in propagation?
HASSAYOUNE	Ahmed	5	Good!
KHAREF	Okba	4.5	Good, though during propagation you should simply not compute NCC for disparities outside [dmin,dmax] instead of thresholding afterwards.
KILINC	Sena	4	For the dense case, pixels with no valid patch still get disparity dmin. During propagation, overflowing dmax should not be allowed.
LAVAL	Luka	4	Initializing bestNcc to 0 is a bad idea, should be -1 or less. Moreover, leaving bestDisparity uninitialized puts random values to pixels having no valid patch.
MAOUCHE	Mounir	3	No propagation as nccSeed should not be reused. Beware of not going out of [dmin,dmax]. For dense map, pixels with no valid patch should not get dmin.
OUMAZIZ	Thiziri	4	In the dense estimation, pixels with no valid corresponding patch still get dmin, which is an admissible disparity. They should not. In propagation, bestNCC is initialized to -1 and can only increase, so the check ≥ -1.0 is useless and pixels with no valid patch still get assigned the disparity of the seed.
PAGNEUX	Gabriel	3	In find_seeds, pixels with no valid patch for all disparities get assigned a random (uninitialized) value. Moreover, initial value of ncc_best must be -1 or less. Propagation does not happen because by definition non-seeds have NCC less than nccSeed.
RCHAKI	Oussama	4	Pixels with no suitable patch still get assigned a disparity, which is not correct. Check overflow of patch to the right also during propagation. $W=2$ should not be tested.
ROUSSELLE	Naomi	4.5	Propagation should not let disparity outside [dmin,dmax]. In find_seeds, if $dmin \leq -1 \leq dmax$, pixels with no valid patch would still get assigned a disparity.
SELLAHENNEDI	Ménalie	1.5	Results are completely wrong, due to several bugs: in correl p2 is shifted by m1 instead of m2, the +EPS should be in denominator. In the dense case, some pixels with no valid patch get a random disparity. Why the threshold 0 for NCC in propagate?
SHIKHLI	Nadir	5	Very good!
SIDKI	Noureddine	4	In find_seeds, pixels with negative best NCC have no disparity, which leaves holes in dense configuration. Valid patch in image 2 should compare to image2.width, not image1. The propagation does not check we stay within [dmin,dmax].
SOARES	Robin	4.5	In the dense case, the pixels having no valid corresponding patch get assigned a random (uninitialized) disparity. OK for propagation.
TIAN	Ning	4.5	Good, but during propagation pixels with no valid patch should not get assigned a disparity within [dmin,dmax].

GCDisparity

GCDisparity(/5)

MVA

BELLANCA	Ugo	4.5	Clean code, but for invalid patches w is used while uninitialized: infinity weight should be used in that case.
FOURREAU	Félix	3	Out-of-bounds patches must be detected and $zncc$ not called in this case. Disparity d must not be multiplied by zoom. Disparity d_{max} is never reached in <code>decode_graph</code> .
KADDAMI	Yassine	4	Good code, though <code>decode_graph</code> is more complex than needed. This assignment deserved a report.
MEGHRAOUI	Abdallah	4.5	Nice code, but it is better to have no reverse edges with weight infinity between d and $d-1$: add rather K to direct connection. A report was required for this assignment.
MICHETTI	Marius	4	You are not handling well pixels that have some but not all translated patches valid. In the report, it would have been good to include visual results of the different experiments.
RAVAUD	Tom	5	Very good, both code and report. The only defect is that d_{max} cannot be set in <code>decode_graph</code> .
SAULNIER	François		
SEREYJOL-GARROS	Nicolas	4	Good overall, but $zncc$ must not be called without checking the patch is inside the image bounds.
SOUCAILLE	Matti	3.5	You do not handle patches out of bounds: calling $zncc$ without security check is dangerous. It would have been good to have a report for this assignment.
VERSINI	Lucas	5	Very good, though in <code>decode_graph</code> value d_{max} is not reachable. You could also put 0 as reverse weight between d and $d+1$ instead of infinity, leading to fewer arcs and helping the GC to run a bit faster.
VICTOR	Ludvig	5	Good job!
WAUQUIEZ	Mathis	5	Very good!

IMA

CADI	Yacine	1	The built graph is not correct: intermediate disparity nodes are not connected to the terminals, but to the node with preceding disparity. You did not understand the role of zoom. The tested patch must be at $u+d$, not $u-d$. In <code>decode_graph</code> , the while loop may overflow the array the nodes.
DRACEA	Iulian-Ilie	5	Very good, even though disparity d_{max} gets excluded because $nd-1$ means $d_{max}-1$
DROZDOV	Dmitrii	4.5	You are not taking exactly the weight function suggested, which included a square root. The weight to sink is not right since it should involve disparity d_{max} . Good for the benchmark.
GONCALVES	Samuel	4.5	You are not taking into account the zoom factor. For node (i,j) , the patch would be centered at $(i,j)*zoom+win$. Good report.

GCDisparity

GRUSS	Carlos	4	Function <code>zncc</code> must not be called when the patch is not fully in image. The function <code>decode_graph</code> never yields <code>dmax</code> .
HASSAYOUNE	Ahmed	4	If you add weights to 4 neighbors, each edge will have twice the normal weight because the "neighbor" relation is symmetric. The test of in-bound patch must also check at the left boundary. When comparing runtime with region growing method, you should realize that <code>graph cuts</code> was considerably faster because of zoom factor.
KHAREF	Okba	4.5	Good, but in <code>decode_graph</code> <code>dmax</code> is not reachable and disparity must be reset to <code>dmin</code> for each pixel.
KILINC	Sena	4	If you add weights for the 4 neighbors, you are doubling <code>lambda</code> ; you should consider one horizontal and one vertical neighbors. In <code>decode_graph</code> , <code>dmax+dmin</code> is not a correct disparity.
LAVAL	Luka	4	You are lucky the images are square, since <code>get_id</code> has a bug: <code>x*nx</code> instead of <code>x*ny</code> . When connecting to neighbors, only two must be considered, since the relationship is symmetric. Weights to neighbors at same disparity should be <code>lambda</code> , not 1. Terminal weights are not correct. When comparing runtime, you should not forget that <code>zoom=2</code> means that only 1 over 4 pixels is considered.
MAOUCHE	Mounir	4	Overflow of patch can also occur on the left and should be checked. Disparity <code>dmax</code> cannot be assigned in <code>decode_graph</code> . A report for this assignment is lacking.
OUMAZIZ	Thiziri	0	not handed in.
PAGNEUX	Gabriel	1.5	Since you turned the images, they are no longer horizontally rectified, but vertically! <code>Dmax</code> is not reachable in <code>decode_graph</code> .
RCHAKI	Oussama	4	<code>d</code> must not be multiplied by <code>zoom</code> in <code>targetX</code> . Because of the symmetry of neighbor relationship, you should not add <code>lambda</code> to the 4 neighbors but only 2.
ROUSSELLE	Naomi	4.5	If adding weights for the 4 neighbors, it is as if you doubled <code>lambda</code> since <code>p~q</code> implies <code>q~p</code> . It would have been good to compare empirically with the seeds method.
SELLAHENNEDI	Ménalie	5	Very good!
SHIKHLI	Nadir	4.5	Overflow of patch on the right should also be checked. Disparity <code>dmax</code> cannot be assigned in <code>decode_graph</code> as maximum <code>d</code> is <code>nd-1</code> , not <code>nd</code> . Taking 4 neighbors instead of 2 adds twice <code>lambda</code> to edges of neighboring pixels at same disparity.
SIDKI	Noureddine	4.5	Good, but adding <code>lambda</code> to 4 neighbor amounts to doubling <code>lambda</code> , since the relationship is symmetric. Pixels that must have disparity <code>dmin</code> actually get <code>dmin-1</code> and <code>dmax</code> cannot be reached.
SOARES	Robin	4	The NCC with disparity <code>d</code> must be used for weight to node <code>(x,y,d-1)</code> , not <code>(x,y,d+1)</code> . The loop in <code>decode_graph</code> may overflow the nodes.
TIAN	Ning	4	It is not a great idea to skip disparities with invalid patches as it creates for such pixels a chain that is not connected to one terminal. By putting INF to sink, you are forbidding disparity <code>dmax</code> . A report was required for this assignment.