

## 3DCV 2023

		Grade	TP1	TP2	TP3	TP4	Mean TP(/5)	Exam
<b>MVA</b>								
BAILLET	Victor	16.5	4.5	4.5	4.5	5	4.6	13.5
BORNE-PONS	Paul	14	4.5	5	5	4.5	4.8	7
DE SENNEVILLE	Adh�mar	12.5	5	3.5	4.5	4.5	4.4	5
EVE	C�lestin	15.5	5	3	4	5	4.3	13.5
GAUILLARD	Emma	13.5	4.5	4.5	5	4	4.5	6.5
GOLLIER	Maxence	13	4.5	5	2	3	3.6	11
JANVIER	Maya	14	4.5	4.5	5	4	4.5	8
JAUMAIN	Pierre	12	3	5	5	4	4.3	5
LABIAD	Ismail	14.5	4	4	4.5	5	4.4	10.5
LAFENETRE	Jamy	14.5	4	4	4.5	4.5	4.3	11
MARENGO	Matteo	14	5	4.5	4.5	4	4.5	8.5
MELO	S�bastien	15	5	2	4	4.5	3.9	14
POLLET	Florent	13.5	5	4.5	4	4	4.4	8
RANJBAR	Mahdi		4.5	1.5				
ROBERT	Hugo	13.5	5	4.5	5	4	4.6	5.5
TERRISSE	Theilo	16	5	5	5	5	5	9.5

		Grade	TP1	TP2	TP3	TP4
<b>IMA</b>						
ARAOUI	Mohammed	18	5	4.5	4	4.5
BIENVENU	Marie	16.5	5	4.5	4.5	2.5
BIOSCA MARRON	Pau	16	4	4.5	4.5	3
CHEN	Yunqi	10.5	4.5	4.5	0	1.5
FODIL	Zine-Eddine	18	4.5	4.5	4	5
GUO	Youheng	18	4.5	4.5	4.5	4.5
LEONELLI	Caterina	14.5	4.5	1	4	5
LIU	Wenda	13.5	4.5	4.5	3	1.5
MADRAU	Maxime	14.5	3	4	3.5	4
MELLUL	Yona	19.5	5	4.5	5	5
MOEINI	Mina	12	4	2	4.5	1.5
MUSTROPH	Henryk Alexander	5.5	2	1.5	1	1
OMIDI	Mojan	18	4.5	4.5	4	5
PORTE	Leo	12	5	1.5	3.5	2
RICHARD	Tom	8	3	1	3	1

## 3DCV 2023

RIFFAUD	Hugo	16.5	5	4.5	3	4
SADREDDINI	Kimia	12	4	4.5	0	3.5
SCHADE	Oskar Hartvig	10.5	3.5	1.5	4.5	1
THOMAS	Guillaume	6.5	2.5	1	2	1
XI	Yuxuan	17.5	4.5	4.5	4	4.5
YANG	Kai	13	4.5	0	4	4.5
ZHANG	Jacques	9	4.5	2	2.5	0

Auditors		Grade	TP1	TP2	TP3	TP4	Mean TP(/5)	Exam
ADJEVI-NEGLOKPE	Ambre	12.5	3	4	3.5	2		4.5
ZEINATY	Paul	18	5	5	3.5	4.5		

Panorama

Panorama(/5)

MVA

BAILLET	Victor	4.5	Good work overall, but careful, 3 components for Color so at line 27 the loop condition must be $I < 3$ , not $I < 4$ .
BORNE-PONS	Paul	4.5	see comments for Emma Gaillard. Good for adding Doxygen comments.
DE SENNEVILLE	Adhémar	5	Great job!
EVE	Célestin	5	Good, code clean and tidy.
GAUILLARD	Emma	4.5	When pulling, the translation by $(x_0, y_0)$ must be applied before the homography, not after. Do not recompute inverse of H at each iteration. Detecting overlap with white color is not the best idea.
GOLLIER	Maxence	4.5	Everything good, except the missing transparency at overlapping pixels.
JANVIER	Maya	4.5	see comments for Emma Gaillard.
JAUMAIN	Pierre	3	Disappointing results are due to one main problem: signs of B are wrong (should be +, not -). When fetching pixel in I1, point should be translated before applying the homography, not after. White pixel has values 255, not 0, anyway that is not the best way to check overlap. Do not recompute inverse of H at each pixel. Good for your experimental effort to assess the quality.
LABIAD	Ismail	4	You applied the push method, which yields missing pixels. Moreover detecting overlap through white color is not optimal.
LAFENETRE	Jamy	4	Your panorama looks fine because the upper left corner of I1 coincides with the one of the panorama, but it does not stand in general.
MARENGO	Matteo	5	Good, though the code for blending could have been shorter with factorization.
MELO	Sébastien	5	Good, though the clicking part could have been more friendly with anyGetMouse.
POLLET	Florent	5	Good job!
RANJBAR	Mahdi	4.5	It is somewhat wasteful to recompute the inverse of H at each pixel. Detection of overlap by white pixel is not quite satisfying.
ROBERT	Hugo	5	Good, but don't compute twice $\text{inverseH} * \text{pt}^2$ .
TERRISSE	Theïlo	5	Exactly as requested!

IMA

ARAOUI	Mohammed	5	Very good! However, take care of code presentation (indentation, mutple consecutive blank lines with no logic).
BIENVENU	Marie	5	Very good! For the bilinear interpolation, you could have used directly <code>Image::interpolate</code> .
BIOSCA MARRON	Pau	4	Why is your point normalization after application of H commented out? It has to happen to get correct pixel coordinates. You did not implement transparency.
CHEN	Yunqi	4.5	Your transparency application in two steps could be simplified.

## Panorama

FODIL	Zine-Eddine	4.5	The color distortion in the transparency is due to an overflow of color components, stored in unsigned char.
GUO	Youheng	4.5	Transparency is missing. Compute the inverse of H outside the loop.
LEONELLI	Caterina	4.5	The transparency on overlapping regions is missing, fine otherwise.
LIU	Wenda	4.5	Your way of applying the transparency is convoluted, it could be done directly.
MADRAU	Maxime	3	Results are not very good because you have some wrong values in matrix A: $A(2i,7)=-y1*x2$ and $A(2i+1,6)=-x1*y2$ . Moreover, you forget to normalize by the third component when multiplying by the homography to retrieve pixel coordinates.
MELLUL	Yona	5	Good! NB: on a MacBook, clicking with two fingers simulates a right click.
MOEINI	Mina	4	Please send the source code next time, not only the report. What is missing is the transparency
MUSTROPH	Henryk Alexandre	2	Something is wrong with the clicks, last points look suspect. In the panorama construction, you check if the point is inside image with the dimensions of I instead of I1 and I2. You do not implement the transparency.
OMIDI	Mojan	4.5	There is no blending or transparency, which would have helped assess the quality of the panorama. However, very good for trying on your own data.
PORTE	Leo	5	Very good!
RICHARD	Tom	3	To pull pixels from I1, you have to apply the inverse of H. Moreover, a translation by (x0,y0) must occur before, not after. Transparency is not handled correctly, white pixel is (255,255,255).
RIFFAUD	Hugo	5	Great, and congrats for having tested on your own data!
SADREDDINI	Kimia	4	Good, except the transparency is not implemented.
SCHADE	Oskar Hartvig	3.5	You did not get the interest of anyGetMouse, which indicates in which window the click occurred. To pull pixels from I2, you have to apply H, not its inverse. However, the bounding box was computed assuming I2 would stay fixed and I1 mapped.
THOMAS	Guillaume	2.5	The homography should be applied to a single image, the other has just a translation. Moreover, to get image coordinates, it is necessary to divide by the 3 <sup>rd</sup> component.
XI	Yuxuan	4.5	You did not apply the transparency effect, but otherwise it is fine.
YANG	Kai	4.5	Good, only the transparency on overlapping regions is missing.
ZHANG	Jacques	4.5	Good for testing on your own data. The number of clicked points should not be fixed in the program, right click should stop. Transparency is not implemented. (0,0) is a valid pixel.

Auditors

## Panorama

ADJEVI-NEGLOKPE	Ambre	3	You should not expect unfilled coefficients of matrices to be zero. The code assumes that point (0,0) of I1 and panorama coincide. What you do is push pixels from I2 to the panorama, hence the missing pixels. Transparency has an overflow because channels are unsigned char and the addition may wrap around 0.
ZEINATY	Paul	5	Good, but some feedback would have been welcome while clicking.

Fundamental

Fundamental(/5)

MVA

BAILLET	Victor	4.5	LS refinement is missing at the end. With thin SVD and A of size 8x9, we should have Vt of size 8x9, and the last column missing. Beware, $10e-3=1e-2=0.01$ .
BORNE-PONS	Paul	5	Very good, but the two functions for 8-point and n-point algorithms could be merged.
DE SENNEVILLE	Adhémar	3.5	Normalizing the matches inplace prevents their display in main function. You should not compute the SVD of $A^t A$ but the one of A. In the refinement, you leave the last row of A undefined because of a loop bound off by 1. The user interaction is cumbersome because two clicks are expected. Good for trying on custom data.
EVE	Célestin	3	The vector inliers is not cleared at each iteration, in some runs I had more inliers than matches! No LS refinement at the end.
GAUILLARD	Emma	4.5	By default, the svd function should compute the thin SVD, hence Vt would be 8x9 and the last row is missing. Because of a bug, it computes the full SVD, hence you are spared. Niter may become -infy if $1-(m/n)^8=1$ (too few inliers).
GOLLIER	Maxence	5	Very good, though the fixed min number of inliers to catch the numerical problem could be better handled.
JANVIER	Maya	4.5	You should be careful, unset coefficients of Vector and Matrix are not initialized and may have random values. The formula for Niter update may suffer from numerical problem when the proportion of inliers is small (typically less than 10%).
JAUMAIN	Pierre	5	Good, though repeating very similar code for the minimal solver and the LS is not great.
LABIAD	Ismail	4	You should not compute $A^t A$ , the SVD of A has the same V and it is numerically more favorable. After RANSAC, a least square minimization should be performed.
LAFENETRE	Jamy	4	This is good work, unfortunately results are disappointing due to a dumb mistake: y2 is computed from matches_sample[j].y1. Niter needs an update only when the number of inliers increases, not at each iteration. Very good for the gradient descent (which is not stochastic, by the way), but would be better with parameterization by the normalized F matrix.
MARENGO	Matteo	4.5	There can be a numerical problem with $1-(m/n)^8=1$ , hence a division by 0 in the formula for the update of Niter. The minimal solver and LS could have been merged.
MELO	Sébastien	2	Bad results because the vectors u and v have their last component uninitialized (should be 1) and some coefficients of N are left also uninitialized. Also $10e-3=0.01$ , you meant $1e-3$ , which is 0.001. Applying blindly the formula for Niter may lead to division by 0 because of numerical problems.
POLLET	Florent	4.5	Nothing wrong with your code, however Imagine++'s doc says that by default svd computes the thin SVD, hence if A is 8x9, Vt should be 8x9 and the last column of V would be missing. It happens there is a bug and the full SVD is always computed.

Fundamental

RANJBAR	Mahdi	1.5	You don't compute the denominator for the point-line distance, and at the end compare with the squared maximum distance. The number of inliers is completely wrong (I got all matches).
ROBERT	Hugo	4.5	Good! However, you should not compute $A^t A$ , the SVD of A is enough and numerically better.
TERRISSE	Theïlo	5	Very good!

IMA

ARAOUI	Mohammed	4.5	Good but the update formula for Niter should be protected from numerical problems (potential division by 0). Indentation of the code is disastrous. Notice it makes no sense to try on panorama images, there is no translation.
BIENVENU	Marie	4.5	Very good, except the formula for update of Niter must be protected from numerical errors. And you inverted F and its transpose in <code>displayEpipolar!</code>
BIOSCA MARRON	Pau	4.5	Very good! LS estimation should still be performed after RANSAC and the formula for Niter update but be protected from numerical approximations.
CHEN	Yunqi	4.5	Very good, but refinement has a problem: the loop for filling A uses only the first 8 points.
FODIL	Zine-Eddine	4.5	No LS refinement. Moreover, if some model has too few inliers, you can get $1-(m/n)^8=1$ and a division by 0, hence Niter=-infy and the loop stops.
GUO	Youheng	4.5	Good, would have been even better with least square refinement using all inliers.
LEONELLI	Caterina	1	Plagiarism Sadreddini for the code. Fortunately, you provided good experimental evaluation after my complaint.
LIU	Wenda	4.5	Missing LS at the end of RANSAC. Good otherwise, though the code is a bit verbose.
MADRAU	Maxime	4	No LS after RANSAC. The formula for Niter must be applied with precautions. When clicking on left image, the y of right point is wrong since its logical x is w, not 2*w.
MELLUL	Yona	4.5	The LS refinement is not applied, and should actually involve all inliers, not just 9. Off-diagonal coefficients of N are not initialized.
MOEINI	Mina	2	Two fatal bugs: (1) <code>line[2]</code> is missing in formula for point-line distance. (2) Two counters, <code>samplingN</code> and <code>samplingNb</code> , one is incremented but the other is tested.
MUSTROPH	Henryk Alexandre	1.5	Your normalization of points makes no sense, they should be all normalized with the same factor. <code>A(8,8)</code> is left uninitialized. What is the role of <code>x_values</code> and <code>y_values</code> (which are uninitialized, by the way)?
OMIDI	Mojan	4.5	Good job! The only missing part is the LS refinement. Also, your threshold at 50% inliers for update of Niter is too conservative.
PORTE	Leo	1.5	<code>V(V.nrow(),...)</code> is always out of bounds, you mean <code>V(V.nrow()-1,...)</code> . Anyway, you apply thin SVD and V is 9x8, <code>V9</code> is missing. If you compute line-point distance in normalized space, the threshold <code>distMax</code> must be adapted.

### Fundamental

RICHARD	Tom	1	There is a confusion with H computation. Here, matrix A is 8x9 and each match gives a single equation.
RIFFAUD	Hugo	4.5	Good, only LS estimation is missing at the end of RANSAC.
SADREDDINI	Kimia	4.5	Good! What is missing is the LS refinement with all inliers. The safety against numerical problem in Niter update is not flexible with its fixed threshold.
SCHADE	Oskar Hartvig	1.5	When refining the estimation with all inliers, matrix A has the wrong size and does not require the addition of a line of zeros. There is no enforcement of rank-2 constraint. When drawing lines, line should be displayed on right part when click on left part, and thickness is wrong.
THOMAS	Guillaume	1	Confusient with computation of H. For F, each match gives a single equation, hence a single row of matrix A.
XI	Yuxuan	4.5	Missing LS refinement at the end of RANSAC and lack of precaution when applying the update formula for Niter.
YANG	Kai	0	Plagiarism
ZHANG	Jacques	2	Several bugs: $A(i,2)=x1$ , not $x2$ . Off-diagonal coefficients of N not initialized. Reshaping Vt as F actually computes its transpose.

### Auditors

ADJEVI-NEGLOKPE	Ambre	4	If A is 8x9, Vt should be also 8x9 if thin SVD is applied (default for function svd), and the last column of V is missing. Beware, $10e-3=0.01$ . The code for displayEpipolar is hardly readable and $i\_min1$ is left uninitialized. Precautions should be taken when applying the update for Niter.
ZEINATY	Paul	5	Very good!



Seeds

Seeds(/5)

MVA

BAILLET	Victor	4.5	bestDisparity may be assigned uninitialized to pixels. It happens at the left border, where every patch overflows the image.
BORNE-PONS	Paul	5	Very good!
DE SENNEVILLE	Adh�mar	4.5	Good report! d_end should be excluded from the loop, as it would overflow if dmax were positive. In propagate, some pixels are assigned disparity 0 because they have no valid patch. Clipping the best disparity is not as good as just ignoring invalid disparities.
EVE	C�lestin	4	In find_seeds, pixels with no valid patch are assigned dBEST, which is uninitialized. Same with ddBest in propagate. The patch may also overflow to the right if dmax were positive. No check of disparity bounds during propagation.
GAUILLARD	Emma	5	Very good!
GOLLIER	Maxence	2	Results are particularly bad due to a dumb bug: in correl, denominator is sqrt(var1)*sart(var2), not with pixel1 and pixel2! Pixels with no valid patch get assigned an uninitialized disparity. Right overflow of patch should also be taken care of. During propagation, some pixels have no valid patch and should not get a disparity. Clamping to [dmin,dmax] is worse than ignoring bad disparities.
JANVIER	Maya	5	Very good!
JAUMAIN	Pierre	5	Very good work! Just a little problem in propagate, where uninitialized bestDisparity is assigned to pixels with zero valid patch.
LABIAD	Ismail	4.5	During propagation, clamping to [dmin,dmax] is not as good as just ignoring wrong disparities. Moreover, it assigns dmin to pixels that have no valid patch. Notice also that the right overflow check should be stricter with >=im2.width(), not >.
LAFENETRE	Jamy	4.5	Good. No check for disparity bounds during propagation. The median filter does not help that much as wrong pixels are not initial seeds and are bound to have close disparity as their neighbors.
MARENGO	Matteo	4.5	Clamping the disparity to [dmin,dmax] after the minimization is not as good as just ignoring disparities out of range. Overflow of a patch to the right should also be guarded against, since in all generality dmax could be positive (no control over the crop or the rectification).
MELO	S�bastien	4	In find_seeds, pixels on the left with no valid patch are assigned argmaxd, which is uninitialized. The same for pixels whose best NCC is negative. Overflow on the right should be checked with im2.width() (not im1), and the inequality should be >=. No check of disparity range [dmin,dmax] in propagate. Idem, argmaxd may be assigned while uninitialized.

## Seeds

POLLET	Florent	4	During propagation, out of bounds checks must still be performed. Clamping the disparity between [dmin,dmax] is worse than just ignoring wrong disparities. Finally, some pixels may have no valid patch and should be assigned a disparity.
RANJBAR	Mahdi		
ROBERT	Hugo	5	Very good! Still, during propagation, we should accept pixels with bestNCC negative. A better test is <code>bestNCC &gt; -1.0f</code> .
TERRISSE	Theïlo	5	Very good, but the overflow should be checked wrt to <code>im2.width()</code> during propagation, not <code>im1</code> .

## IMA

ARAOUI	Mohammed	4	In <code>find_seeds</code> , initializing <code>Max_ncc</code> as 0 is a mistake, as pixels with no valid patch are still assigned <code>Best_disparity</code> , which is uninitialized. In the propagation, <code>bestDisparity</code> would rather never be tested if outside [dmin,dmax]. It is wasteful to add several times the same seed.
BIENVENU	Marie	4.5	Good work. During propagation, bounds [dmin,dmax] must still be observed. In <code>find_seeds</code> , the checks for bounds are useless, this is already ensured by the bounds of the for loop.
BIOSCA MARRON	Pau	4.5	In <code>find_seeds</code> , check for overflow to the right should be performed, it can happen with <code>dmax &gt;= 0</code> . In propagation, clamping the best disparity to [dmin,dmax] is not the best solution. Some pixels to the left with no valid patch still get <code>bestDisparity</code> , which is <code>dmax</code> due to clamping and <code>dmax &lt; 0</code> .
CHEN	Yunqi	0	Plagiarism of Wenda Liu's work. The few changes introduced a bug: variable <code>dp</code> should vary around <code>s.d</code> , not <code>s.x...</code>
FODIL	Zine-Eddine	4	In <code>find_seeds</code> , you are too conservative: if some disparities but not all in [dmin,dmax] yield a valid patch, they should be tested. Overflow to the right should be checked (could happen if <code>dmax &gt; 0</code> ). During propagation, it is not <code>bestCorr</code> that should be compared to <code>dmin</code> and <code>dmax</code> , but <code>dispar...</code>
GUO	Youheng	4.5	During propagation, clamping to [dmin,dmax] should not be performed after testing the 3 disparities, it is better to discard the ones outside the interval. Also, some pixels with no valid patch are still assigned <code>dmin</code> .
LEONELLI	Caterina	4	During propagation, some pixels have no valid patch associated, they should not be assigned a disparity. It is best to not consider disparities outside [dmin,dmax] then to clamp the result. There could be some overflow of the patch to the right (with different <code>dmax</code> ), this should be checked.
LIU	Wenda	3	In functions <code>correl</code> and <code>sum</code> , you must go to win inclusive, otherwise the normalization in <code>ccorrel</code> is wrong. Moreover, in these functions you implicitly do 0-padding, which produces artificial patches. It is better to not consider overflowing patches. In <code>propagate</code> , variable <code>best_cur</code> must be reset inside the <code>for(i)</code> loop, not before.

## Seeds

MADRAU	Maxime	3.5	In find_seeds, a disparity is still set even though the NCC does not exceed nccSeed, which interferes with visualization of the seeds. During propagation, no check of patch overflow is performed. In non-Release mode, an assert stops the program.
MELLUL	Yona	5	Very good!
MOEINI	Mina	4.5	In propagate, clamping to [dmin,dmax] a posteriori is not the best solution. Some pixels with no valid patch have best uninitialized compared to dmin.
MUSTROPH	Henryk Alexande	1	I think you have not understood at all the principle. For each possible disparity, we compute the NCC and finally select the argument of the maximum.
OMIDI	Mojan	4	In find_seeds, the test is too conservative, some disparities in [dmin,dmax] may be testable with all being so. During the propagation, you test also s.d+2. Clamping the disparity afterwards is not the best. When you test with other images, you must adust dmin and dmax.
PORTE	Leo	3.5	nccSeed has no role in propagation. There should also be a problem because at the bottom and the right you have disparities whereas the patch does not fit within the image.
RICHARD	Tom	3	Disparities should be tested only within the range [dmin,dmax]. A bit too conservative in the propagation, some pixels have one or two valid patches wihin [x+dp-1,x+dp+1]. Fatal mistake in propagate: conditions for if with = instead of ==. Infinite loop if new seeds are not marked during propagation.
RIFFAUD	Hugo	3	Some dumb bugs that ruin the results: (1) in correl, division by $\sqrt{\text{cpt\_im2}} * \sqrt{\text{cpt\_im2}}$ instead of $\sqrt{\text{cpt\_im1}} * \sqrt{\text{cpt\_im2}}$ , (2) in sum, $\text{im}(i+v,j+v)$ instead of $\text{im}(i+u,j+v)$ . Check for overflow of patch to the right should be performed. During propagation, for some pixels none of the three disparities yields an NCC, so best_disparity is compared to dmin without being set.
SADREDDINI	Kimia	0	Plagiarism. This is a copy-paste of a previous year's code.
SCHADE	Oskar Hartvig	4.5	During propagation, no check of disparity within [dmin,dmax] is performed. Also, some pixels with no valid patch still have a disparity assigned. Good otherwise.
THOMAS	Guillaume	2	You are mixing dmax, which is the maximal possible disparity, with an NCC. The range [dmin,dmax] is the set of possible disparities. Overflow on the right would be possible if dMax were positive.
XI	Yuxuan	4	In find_seeds, if some but not all of the disparities in [dmin,dmax] yield a valid patch, they should be tested. Overflow of a patch to the right should also be tested. Clamping to [dmin,dmax] is not the best. Moreover, for pixels with no valid patch, best_d is uninitialized and still compared to dmin and dmax.
YANG	Kai	4	In find_seeds, pixels with no valid patch are still assigned a disparity, which is actually bestD, left uninitialized. Similar problem in propagate. Patch inclusion should also check the right boundary.

### Seeds

ZHANG	Jacques	2.5	nccSeed is unused in find_seeds. In the propagation, why is $\text{disps} > \text{s.d}$ significant? We test $\text{s.d}-1$ and $\text{s.d}$ , which are perfectly fine. When you test on other images, $\text{dMin}$ and $\text{dMax}$ must be adjusted.
-------	---------	-----	---

### Auditors

ADJEVI-NEGLOKPE	Ambre	3.5	Why divide the grey levels by 256 (through and $\text{RGB} < \text{float} >$ on top of that)? NCC is invariant to such a factor. Disparities should be restricted to the interval $[\text{dmin}, \text{dmax}]$ . In propagate, $\text{x} + \text{d} + \text{win}$ should be compared to $\text{im2.width}()$ , not $\text{im1}$ .
ZEINATY	Paul	3.5	In find_seeds, some pixels have their best NCC negative. In that case, since $\text{bestNcc}$ is 0, they are assigned $\text{bestDisp}$ , which is whatever happened before. Moreover, the right overflow check should be $\text{x} + \text{d} >= \text{im2.width}() - \text{win}$ . In propagate, similar problem and the interval $[\text{dmin}, \text{dmax}]$ is not used.

## GCDisparity

GCDisparity(/5)

MVA

BAILLET	Victor	5	Great job!
BORNE-PONS	Paul	4.5	Good, though you add lambda twice by using 4 neighbors instead of 2. By putting INF as terminal weights, you are excluding dmin and dmax. Zncc at disparity d should feed the weight from d-1 to d, not d to d+1. Great gif animations!
DE SENNEVILLE	Adhémar	4.5	You forgot to put weight to y+1, hence the apparent horizontal stripes in the results. By putting infinite weight to sink, you are forbidding disparity dmax, which should actually depend on zncc. If you authorize it back, decode_graph should be adapted to it. Great report!
EVE	Célestin	5	Very good!
GAUILLARD	Emma	4	Disparities dmin and dmax are excluded as you put infinite weights to terminal nodes. Using the four neighbors amounts to adding twice lambda.
GOLLIER	Maxence	3	In call to zncc, disparity must not be multiplied by zoom as it is computed in original images. Put INF as weight if the patch is not in the image. In decode_graph, the result should be dmin+d, no normalization has to be performed.
JANVIER	Maya	4	Putting (INF,0) and (0,INF) as terminal weights excludes disparities dmin and dmax. ZNCC at dmin+d should be used with nodes at d-1 and d, not d and d+1. If you consider 4 neighbors, you are adding twice lambda to the edges. In decode_graph, the test $d < nd$ should precede the call to what_segment. Good report.
JAUMAIN	Pierre	4	You are forbidding dmin and dmax as disparity because you put infinite weight from the source and to the sink. All is fine otherwise. The vertical streaks in the result are due to adding twice lambda for vertical smoothness but no horizontal smoothness.
LABIAD	Ismail	5	Excellent, both code and report!
LAFENETRE	Jamy	4.5	Very good, but considering the 4 neighbors amounts to adding lambda twice because (x,x+1) are neighbors and (x+1,x) also.
MARENGO	Matteo	4	All calls to zncc must be preceded by checks that the patch is within the image. The weight to sink should depend on zncc with disparity dmax. decode_graph does not detect correctly disparity dmax (all nodes should be linked to the source). Good report.
MELO	Sébastien	4.5	Good, just a minor mistake: you are still using the zncc at dmax-1 to feed the weight to sink.
POLLET	Florent	4	Function zncc cannot be called before checking the patch is within the image. In Release mode, assertions are disabled and it can go unnoticed. The weight for d=0 should correspond to disparity dmin, not dmin+1. Good report.
RANJBAR	Mahdi		
ROBERT	Hugo	4	You cannot call zncc without guarding against patch overflow (assertion triggered in non-Release mode). In decode_graph, you would rather put dmax by default, as if the cut happens just before the sink, this indicates dmax. Very good report!

GCDisparity

TERRISSE	Theïlo	5	Excellent, clean code and nice report!
----------	--------	---	--

IMA

ARAOUI	Mohammed	4.5	Good, but zncc at d should feed weight from (d-1) to d, not d to (d+1). Problem in decode_graph if the disparity is actually dmax.
BIENVENU	Marie	2.5	There is some confusion about indices which are incremented by 1. This being fixed, decode_graph should add dmin and put dmax to unassigned pixels.
BIOSCA MARRON	Pau	3	You do not add K. Terminal weights are wrong. Decoding should put dmin+d.
CHEN	Yunqi	1.5	It does not work because you put INF weights to terminal links, which prevents cutting them.
FODIL	Zine-Eddine	5	Very good! Still, you should not rely on $d \geq 0$ and ignore the possibility of left overflow of patch. Beware that computation time is with zoom=2.
GUO	Youheng	4.5	Good overall, but you forget to add K to the cost, overflow check should not multiply $d+dmin$ by zoom and weight to sink should depend on ZNCC at dmax, not $dmax-1$ . Good! However, you should put weight INF instead of wcc for overflowing patch.
LEONELLI	Caterina	5	When testing on kitchen images, results cannot be good if the images are not rectified.
LIU	Wenda	1.5	Your weights to terminals are wrong. Why put $(nd+20)*wcc$ ?
MADRAU	Maxime	4	The code must not call zncc unguarded against patch overflow, which happens here. Otherwise, everything fine.
MELLUL	Yona	5	Very good! Do not ignore the possibility of left overflow in case $d < 0$ .
MOEINI	Mina	1.5	You cannot compute zncc without checking the patch is within the image. In decode_graph, you must put $D=dmin+d$ , the integer disparity.
MUSTROPH	Henryk Alexandre	1	Your node-number formula is wrong, $x*y$ should not appear inside. You do not use ZNCC as unary weights. Why do you have a double for loop on depth?
OMIDI	Mojan	5	Very good! When testing on other images, dmin and dmax must be adjusted.
PORTE	Leo	2	You forgot to add K to the weight, which changes everything. When decoding, you must add dmin. Your function rho is not exactly the proposed one.
RICHARD	Tom	1	Everything should be changed: node numbers are wrong, weight does not care of zoom factor, decoding should just assign d.
RIFFAUD	Hugo	4	ZNCC with d must be used to link edge d-1 to d, not d to d+1. In decode_graph, result should be dmin+d, not d.
SADREDDINI	Kimia	3.5	Terminal weights are wrong: should be (w,0) for $d=0$ and (0,w) for $d=nd-1$ . You do not add K. Weight should be INF for out-of-bounds patch.
SCHADE	Oskar Hartvig	1	You did not understand the role of zoom: $n_x$ and $n_y$ already take into account the zoom factor and the window radius. decode_graphs should be fine except that dmax must be excluded as disparity dmax is encoded by the cut occurring just before the sink.

### GCDisparity

THOMAS	Guillaume	1	In build_graph, x and y should start from 0 and stop at nx and ny, win is already taken care of. Terminal node weights are wrong, do not put infinity. neighbor_node_id is wrong, should be with d-1. Moreover, the zncc must be mapped through the function $\sqrt{1-x}$ as weight.
XI	Yuxuan	4.5	The weight to the sink should depend on zncc at $d_{max}=d+nd$ , not $d+nd-1$ . Very good otherwise. When comparing on other images, $d_{min}$ and $d_{max}$ must be adjusted.
YANG	Kai	4.5	You should add K to the weights. You should check also overflow of patch by the left side (in case $d_{min}<0$ ). Default value for C in build_graph should be INF.
ZHANG	Jacques		

### Auditors

ADJEVI-NEGLOKPE	Ambre	2	You do not take into account that there is a zoom factor to speed up the computation, so the real x and y are $x*zoom+win$ and $y*zoom+win$ . Calls to zncc must be done only after checking the patch is within the image.
ZEINATY	Paul	4.5	Good! Still, the weight of $(nd-1)$ to the sink should depend on zncc at disparity $d_{max}$ , not $d_{max}-1$ . The x and y increments are not very intuitive. Good report.