Take-home Midterm Examination

CS5670: Computer Vision, Cornell Tech, Spring 2021

Instructor: Noah Snavely
TAs: Ruojin Cai, Zikai Alex Wen, and Qianqian Wang March 15, 2021

Assigned: Monday, May 15, 2020, 4:15pm EDT
Due: Friday, May 19, 2020, 7pm EDT (via Gradescope)

Late exams will not be accepted.

You may complete this exam in the following ways:

1. (Preferred.) You can either (a) print out the exam and scan/photograph it when finished, or (b) complete the exam on a tablet computer with a stylus.
2. If neither approach in (1) works for you, you may instead write up your answers on separate sheets of paper, scan/photograph them, and submit them.

Your scanned/photographed/completed exam should be submitted via Gradescope.

If you are writing on this exam (either physically or digitally), as opposed to on separate sheets of paper please write your name and NetId/email on the top of this page **now**.

This exam has a total of 15 pages, including this one. Please make sure you submit a complete exam.

Please provide answers to the questions in the space provided, or on extra pages that you attach to the exam (or on completely separate sheets, if you are using approach (2) above).

This take-home exam is open book and open notes, but do not discuss it or collaborate with other students. Remember the academic integrity policies (`https://cuinfo.cornell.edu/aic.cfm`). You may refer to the textbook or to the slides online (and to other resources mentioned on the Lectures webpage), but you may not use Google or any other online resource not explicitly allowed. You can use a calculator for normal arithmetic and functions like log, pow, sin, cos—to be exact, you may use the functions available on this online calculator: `http://calculator-1.com/scientific/`.

Make your answers as concise as possible, though remember to show your work. If you feel that a question is unclear, please simply answer the question and state your assumptions clearly.

If you do need to ask the course staff a question, for instance, if you feel there is an error in a question, please create a private post on Ed Discussions (anonymous if you choose). Please do not ask exam-related questions live to the course staff, but instead use private Ed Discussions posts.

Remember to show your work. Good luck!

1. **Image filtering** [16 pts]

   (a) [3 pts] Let a contiguous region of a 2D image **I** be

...	1	5	10	3	8	6	...
...	10	5	5	3	5	6	...
...	3	2	5	2	5	6	...

   Let **F** be a 3x3 kernel:

r	p	r
p	q	p
r	p	r

   (The variables p, q, and r represent values that you will solve for.)

   The result of convolving **I** with **F** (that is, $\mathbf{I} * \mathbf{F}$) gives us the following row of output values (centered in the region of **I** above):

...	3	2	0	3	...

   Solve for p, q, and r. Note that you should ignore any boundary conditions (assume we are very far from the boundary of the image).

   - $p =$ -1
   - $q =$ 5
   - $r =$ 0

(b) [2 pts] What kind of kernel does $\mathbf{F}$ from (a) resemble? Describe in one sentence.

0	-1	0
-1	5	-1
0	-1	0

It resembles edge detection a type of kernel which detects edges within an image.

(c) [3 pts] A 1D smoothing filter of width 3 (appropriate for applying to a 1D image) might look like:

$\frac{1}{4}$	$\frac{1}{2}$	$\frac{1}{4}$

Define a single 1D filter that, when applied only once to an image, will produce the same results as applying the 1D width-3 smoothing filter above twice. Write your filter in the space below.

1/4	1/4	3/2	1/4	1/4

(d) [3 pts] Now moving to 2D, suppose we have the following two filters.

$$\frac{1}{4} \times \begin{array}{|c|c|c|}\hline 0 & 0 & 0 \\ \hline 1 & 2 & 1 \\ \hline 0 & 0 & 0 \\ \hline \end{array}$$

Filter $\mathbf{A}$

$$\frac{1}{4} \times \begin{array}{|c|c|c|}\hline 0 & 1 & 0 \\ \hline 0 & 2 & 0 \\ \hline 0 & 1 & 0 \\ \hline \end{array}$$

Filter $\mathbf{B}$

What is the result of $\mathbf{A} * \mathbf{B}$ (where $*$ denotes convolution)? Write your answer in the space below (all spaces should be filled with some number).

1/16	1/8	1/16
1/8	1/4	1/8
1/16	1/8	1/16

(e) [2 pts] What kind of blur kernel does filter $\mathbf{A} * \mathbf{B}$ resemble?

It resembles 3*3 gaussian blur.

(f) [3 pts] Continuing from (d), given an image $\mathbf{I}$ and the exact kernels $\mathbf{A}$ and $\mathbf{B}$ above, by the associativity of convolution, the results of $(\mathbf{A} * \mathbf{B}) * \mathbf{I}$ are identical to the results of $\mathbf{A} * (\mathbf{B} * \mathbf{I})$. As a practical matter, why might one of these be operations preferable to the other? As a hint, try running both operations by hand, and think about how many additions and multiplications need to be performed in each case, paying attention to optimizations you can perform (e.g., operations you can omit) given the particular values in these kernels.

This saves us from one operation as we can involve image with a single combined kernel instead of performing two derivative operations with respect to image.

2. **Transforms and homographies** [24 pts]

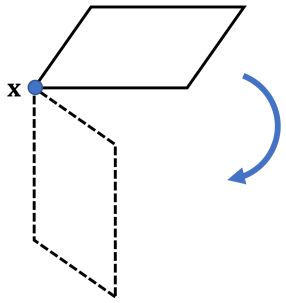
(a) [4 pts] Specify the transformation (as a 3×3 matrix) corresponding to the following image warp: $x' = 3x + 4; y' = 2y - 3$.

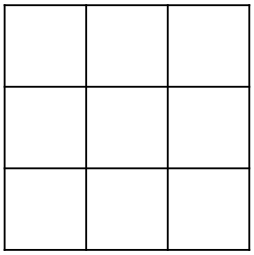
3	0	4
0	2	-3
0	0	1

Which types of transformations occur in this warp? Mark all that apply:

☑ scale ☑ translation ☐ rotation ☐ shear ☐ mirror

(b) [4 pts] Specify the homography corresponding to the following transformation: rotating the parallelogram $90°$ clockwise with $\mathbf{x} = (3, 5)$ fixed. It may be helpful to think of this as a sequence (multiplication) of simpler operations. Show your work.





(c) [3 pts] You are given the following 3×3 transformation matrices:

- $\mathbf{A}$: rotation by θ
- $\mathbf{B}$: rotation by ϕ
- $\mathbf{C}$: mirror reflection about $x = 0$
- $\mathbf{D}$: mirror reflection about $y = 0$

Please specify the transformation matrix for each of the following transformations. You should write these transformations in terms of $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}$ (and their compositions). Recall that the composition of two transformations is given by the product of their corresponding matrices:

- rotation by 2θ:
- rotation by $-\theta$:
- rotation by $\theta - \phi$:

(d) [5 pts] Consider the following three matrices, where the elements a-h are non-zero:

$$\mathbf{A} = \begin{bmatrix} a & 0 & c \\ 0 & e & f \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} a & b & c \\ d & e & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} a & b & c \\ d & e & f \\ h & g & 1 \end{bmatrix}$$

Classify each as one of the following: similarity transformation, affine transformation, homography (choose the most specific class possible, e.g. similarity transformation is more specific than affine transformation):

- $\mathbf{A}$: affine
- $\mathbf{B}$: affine
- $\mathbf{C}$: homography
- $\mathbf{AB}$ (i.e., $\mathbf{A}$ multiplied by $\mathbf{B}$): affine
- $\mathbf{BC}$: homography

(e) [3 pts] Following (e), please specify the minimum number of 2D point correspondences needed to solve for the unknowns in each of the following matrices. (Recall that each 2D point correspondence provides 2 equations in the unknowns):

- **A**:

  x1 = aX1+c
  y1 = eX2+f

- **B**:

  x1 = aX1+bY1+c
  y1 = dX2+eY2

- **C**:

  x1 = a/hX1 + b/gY1 + c
  y1 = d/hX2 + e/gY2 + f

(f) [5 pts] Following (e), for each of the following statements, list all matrices ($\mathbf{A}, \mathbf{B}, \mathbf{C}$) which would preserve the indicated property (there could be multiple matrices per statement):

- angles between lines?

  none

- straight lines remain straight?

  A, B, C

- parallel lines remain parallel?

  A, B

- lengths of line segments?

  none

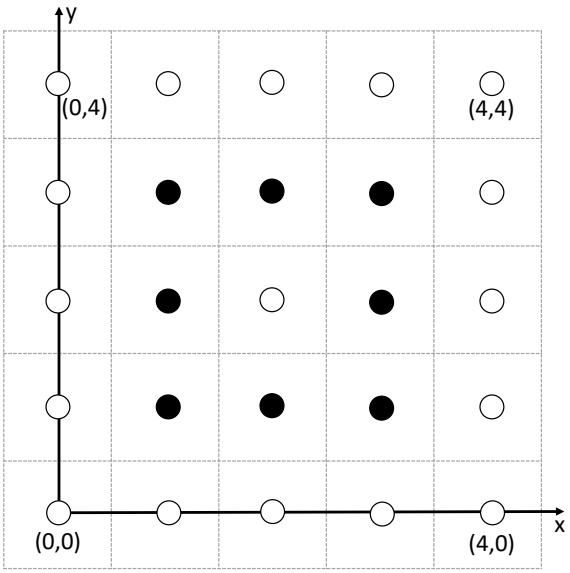
- the ratio between the lengths of two parallel line segments?

  1:1

3. **Resampling and inverse warping** [10 pts]

Below is a 5×5 image f. For white pixels $f(x, y) = 1$, and for the black pixels $f(x, y) = 0$ (i.e., image intensities range from a minimum of 0 to a maximum of 1, and are represented as floating point numbers). Location $(0, 0)$ corresponds to the bottom-left pixel, the x-axis points from left to right, and the y-axis points from bottom to top. We assume pixels outside the image are black (intensity of 0).

For example, the white pixels $f(0, 0) = f(4, 0) = f(2, 2) = f(0, 4) = f(4, 4) = 1$ and the black pixels $f(-1, -1) = f(1, 1) = f(1, 3) = f(3, 1) = f(3, 3) = 0$.



Now we want to compute a transformed image $g(x', y') = f(T(x, y))$, where the coordinate transform $(x', y') = T(x, y)$ is given by the following 3×3 matrix:

$$\begin{bmatrix} 1 & -1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

We want to use inverse warping to assign the intensity of pixel $g(x', y')$ from its corresponding location $(x, y) = T^{-1}(x', y')$ in $f(x, y)$. Note that if pixel comes from "between" pixels, you should interpolate the the intensity value from neighboring pixels.

(a) [4 pts] Please write down the corresponding locations for the listed pixels.

- for the pixel $g(x', y') = g(0, 5)$, its corresponding location in $f(x, y)$ is

$(x, y) =$ (-5/2,5/2)

- for the pixel $g(x', y') = g(-2, 3)$, its corresponding location in $f(x, y)$ is

$(x, y) =$ (-5/2,1/2)

(b) [4 pts] If the source image values are interpolated with a **bilinear** filter, write down the intensity values computed for each listed output pixel. Note that the linear filter is:

$$h_{linear}(x) = \begin{cases} 1 - |x| & |x| < 1 \\ 0 & \text{otherwise} \end{cases},$$

and that bilinear interpolation is performed as described on this page: `https://en.wikipedia.org/wiki/Bilinear_interpolation`.

- the pixel $g(x', y') = g(0, 5) =$ 1

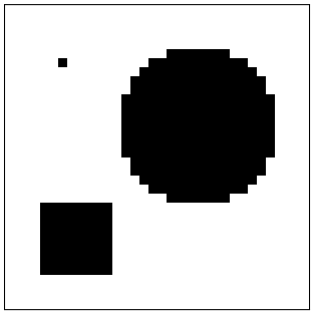
- the pixel $g(x', y') = g(-2, 3) =$ 1

(c) [2 pts] Suppose we compute a Gaussian pyramid from image with size $N \times N$ pixels (where $N = 2^k$ is a power of two). How much space does the Gaussian pyramid take compared to the original image? Write your answer as a multiple of the size of the original image and note that the Gaussian pyramid contains the original image.

$2^{k-l} * 2^{k-l}$ where l = 1.....n

4. **Image filtering II** [6 pts]

Suppose we have the following 32×32 image **I** consisting of black objects on a white background (the image is boundary drawn in black for clarity):



In this problem, we convolve this image with each of a set of convolution kernels, listed below as kernels **A** through **F**. For each resulting filtered image 1-6, specify which kernel produced it as a result of convolving with image **I**, by writing **A**, **B**, **C**, **D**, **E**, or **F** in the space provided next to each image. Note that some of the images have had their intensities shifted such that a value of 0 is gray. All images have been enlarged by a factor of 16 using nearest neighbor upsampling. All convolutions handle image boundaries by replicating the intensities of each boundary row or column outside the input image.

Kernel **A**: 3×3 box filter
Kernel **B**: 3×3 Gaussian filter
Kernel **C**: Sobel filter in x direction
Kernel **D**: Kernel **C** applied to itself
Kernel **E**:

0	−1	0
−1	4	−1
0	−1	0

Kernel **F**:

0.2	0	0	0	0
0	0.2	0	0	0
0	0	0.2	0	0
0	0	0	0.2	0
0	0	0	0	0.2

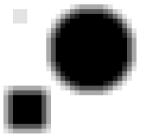
Image 1:  B

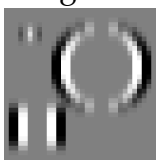
Image 2:  D

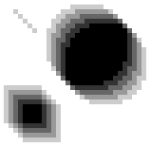
Image 3:  F

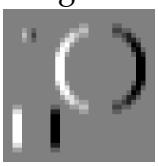
Image 4:  C

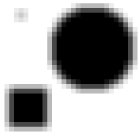
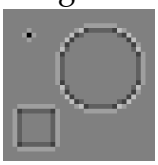
Image 5:  A

Image 6:  E

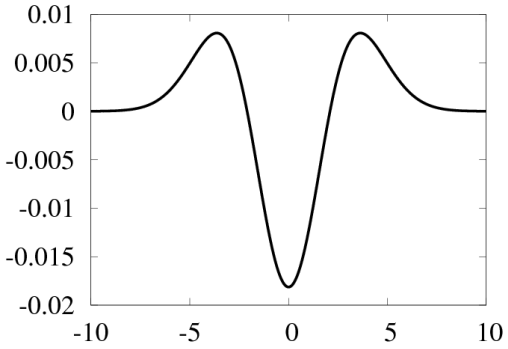
5. **Features** [5 pts]

   (a) [3 pts] Given a continuous image $I(x, y)$, let the function *Harris*$(I; x, y)$ be the Harris operator of I (in this case, assume we take the operator to be the ratio of a product of eigenvalues and a sum of eigenvalues). Consider an image J formed by multiplying I by a constant factor $m > 0$ and adding a constant d:

   $$J(x, y) = mI(x, y) + d$$

   Write *Harris*$(J; x, y)$ in terms of *Harris*$(I; x, y)$. Derive your answer from the definition of the Harris operator (showing your work).

(b) [2 pts] While the Harris corner detector finds features which are maxima of the Harris operator, the SIFT detector finds features that are maxima (or minima) of a different function. SIFT first convolves the images with a *difference of Gaussians* (DoG) filter, which is similar to the Laplacian of Gaussian filter described in class. SIFT then finds maxima and minima of the convolved image. In 1D, the DoG filter looks like this (note that some values are negative):



Consider the three images below labeled Image A, Image B, and Image C. This question asks you to apply the 1D DoG filter to each image at the position $x = 0$, and determine the sign of the resulting number (called the response of the filter). For which images is the filter response at $x = 0$ positive? For which images is it negative? Check one of the boxes for each image; the correct box for Image A has already been checked for you. (The DoG filter has been replicated above each image to help you visualize the convolution). Note that the x-axis of each 1D image represents position, and the y-axis represents intensity (an intensity of 0 indicates black, and an intensity of 1 indicates white).

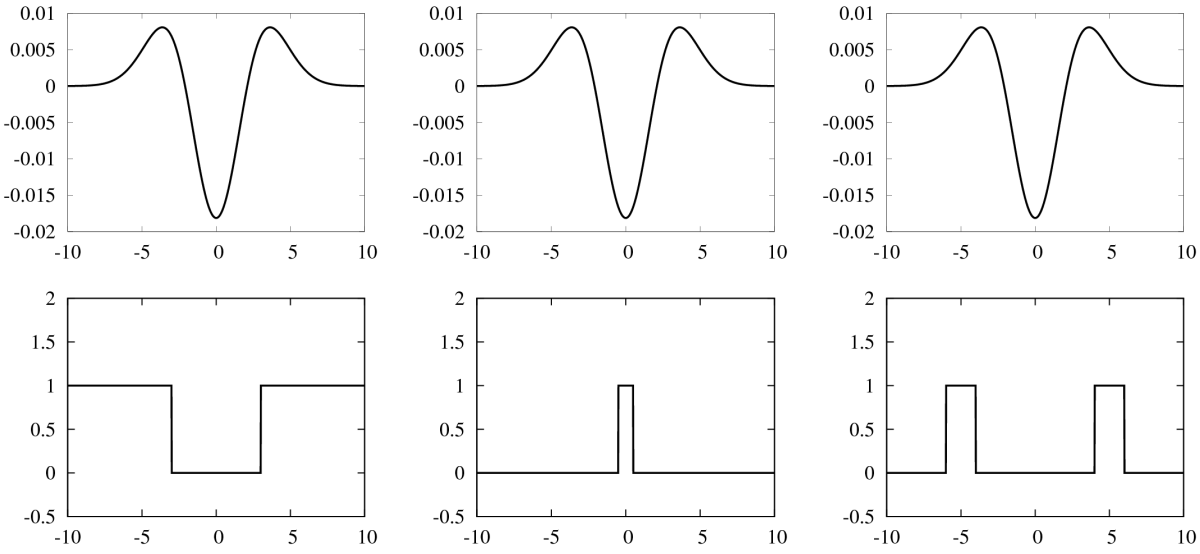


Image A
☒ Positive
☐ Negative

Image B
☐ Positive
☒ Negative

Image C
☒ Positive
☐ Negative

6. **Feature invariance** [5 pts]

   Recall that we can build a "blob" feature detector by convolving an image with a Laplacian of Gaussian kernel (a "second derivative" filter) and looking for local extrema (minima or maxima) of the resulting convolved image. Suppose we do this **at a single scale only**, and not on a Gaussian pyramid as discussed in class. Is this detector invariant/equivariant to the given image operation, i.e., does it detect the same set of points in the image (or, for geometric transformations, the same set of points after applying the same transformation as is applied to the image)? Assume we do not use a threshold (only look for extrema), and ignore issues due to saturation (i.e., pixel intensities going out of range). Circle "yes" (meaning "is invariant/equivariant") or "no" for each operation:

   a. Rotating the image — **Yes** / No

   b. Adding a constant (> 0) to the intensity values — **Yes** / No

   c. Scaling the image to be larger or smaller — Yes / **No**

   d. Squaring the image intensities — **Yes** / No

   e. Multiplying the images intensities by a constant (> 1) — Yes / **No**

7. **Least squares** [6 pts]

   (a) [3 pts] Suppose we want to fit a cubic curve of the form $y = ax^3 + bx^2 + cx + d$ to a set of n 2D data points $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$. Write down a cost function for the corresponding least-squares problem below (i.e., the sum of squared residual errors):

   $$Cost(a, b, c, d) =$$

   (b) [3 pts] Suppose we want to formulate this as a matrix equation $\mathbf{At} = \mathbf{b}$. What would $\mathbf{A}$, $\mathbf{t}$, and $\mathbf{b}$ be in this case? Write down each as a matrix.

8. **Image alignment** [6 pts]

Suppose you have a partially computed 3x3 matrix $\mathbf{H}$ representing a homography between two images:

$$\mathbf{H} = \begin{bmatrix} 2 & 3 & 3 \\ 3 & H_{22} & 1 \\ H_{31} & 1 & 1 \end{bmatrix}$$

where H_{22}, H_{31} represent unknown entries of the matrix.

(a) [4 pts] At least how many additional feature matches would you need to complete the homography, i.e., solve for the missing entries? Why?

To estimate H need to estimate 2 variables. then for this purpose we need to establish the correspondance between at least 1 non alligned point as each point contributes to two equations.

Equations will be like this:
x1 = 2/H31 X1 + 3 Y1 + 3
y1 = 3/H31 X2 + H22 Y2 + 1

(b) [2 pts] Suppose you have three additional feature matches between the two images above:

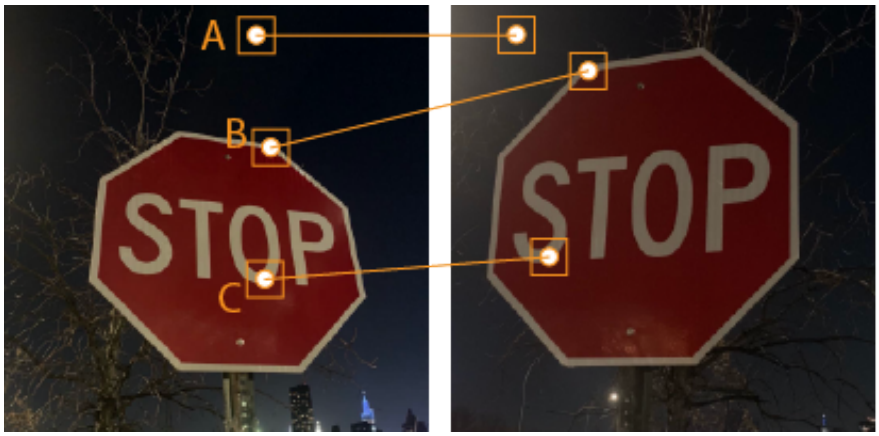
- (1, 0) matches to (5, 4)
- (3, 3) matches to (6, 5)
- (4, 1) matches to (7, 7).

Suppose you know that exactly one of these matches is an outlier (in this case, consider an outlier to be a match that is not precisely correct). Which of the two matches are inliers, and which is an outlier? Explain your answer.

(5,4) and (7,7) are inliners as these points fall within the line of the equations solved from the above provided H. Also able to find fitting lines with these matching points of the image.
While (6,5) is an outliers as it falls outside the line when solved with the following above equations.

9. **Features II** [6 pts]

Consider the two images below. Alex wrote code to match features between these two images. Alex implemented the Harris corner detector, and set the score threshold to 10^{-3}. He used SIFT (with rotation invariance) as the descriptor, and used a SSD (sum of squared differences) matching distance. After matching features between the two images, Alex found that the matches included three incorrect correspondences, labeled A, B, and C.



Alex proposed three modifications to his code. Each modification is designed to eliminate exactly one of the three errors. Please specify which error (i.e., which incorrect match A, B, or C) the modification is trying to address, and briefly explain your reasoning. Each error should be assigned to one fix.

(a) Increase the threshold on the Harris score to accept fewer potential corners

B - should be fixed. By increasing the threshold it finds points ith larger corner responses.

(b) Increase the size of the patch used for the SIFT descriptor

A - should be fixed. As the corner points will be filtered out to give points whose response function is a local maxima by increasing the size of the patch. Then even for the blurriest of the image corner will be detected.

(c) Change from a simple SSD distance to a ratio distance

C - should be fixed. If ratio distance is used d it is smaller than a certain ratio we can establish the correspondance between the specific pair.