Midterm Examination

CS 534: Computational Photography

November 3, 2016

NAME: ______________________________________

Problem	Score	Max Score
1	___________	6
2	___________	8
3	___________	9
4	___________	12
5	___________	4
6	___________	13
7	___________	7
8	___________	6
9	___________	9
10	___________	6
11	___________	14
12	___________	6
Total	___________	100

1. [6]

(a) [3] What camera setting(s) was (were) changed to obtain the following two photographs, with *identical fields of view but different depths of field*? Include in your answer possible values for the setting(s) used for each photo.

(b) [3] If the field of view *can change*, what is another way to *increase* the depth of field?

2. [8]

(a) [3] What happens to the projected ***size*** of an object if the lens' focal length is doubled?

(b) [3] What happens to the ***field of view*** if the focal length is doubled from 50 mm to 100 mm? Give your answer both qualitatively and quantitatively, though giving an equation is enough for your quantitative answer.

(c) [2] True or False: If you take a photo of Abe Lincoln with Bascom Hall in the background, and then take a second photo after doubling the focal length of the camera lens and moving twice as far away from Abe, the two images will be identical.

3. [9] Given a camera with aperture f-number f/4, lens focal length 50 mm, shutter speed 1/125 sec, ISO 400, and focused on an object 500 mm in front of the lens, what is

(a) [3] The *diameter* of the lens?

(b) [3] The *distance* of the image sensor from the lens?

(c) [3] The *height* of the real object if its height in the image is 5 mm?

4. [12] **Gaussian Filters**

(a) [3] Give a 3 x 3 filter that approximates a Gaussian function.

(b) [3] What property of the coefficients of a discrete approximation of a **Gaussian filter** ensures that regions of uniform intensity are unchanged by smoothing using this filter?

(c) [6] Given the 1 x 2 filter: [0.5 0.5]

(i) [3] What is the result of convolving this filter with itself? Give all values that are non-zero, assuming input values of 0 outside of the two given values. Hint: The result will be a 1 x 3 array.

(ii) [3] What is the result of convolving your result from (i) with this same 1 x 2 filter again? Give all non-zero values in the result.

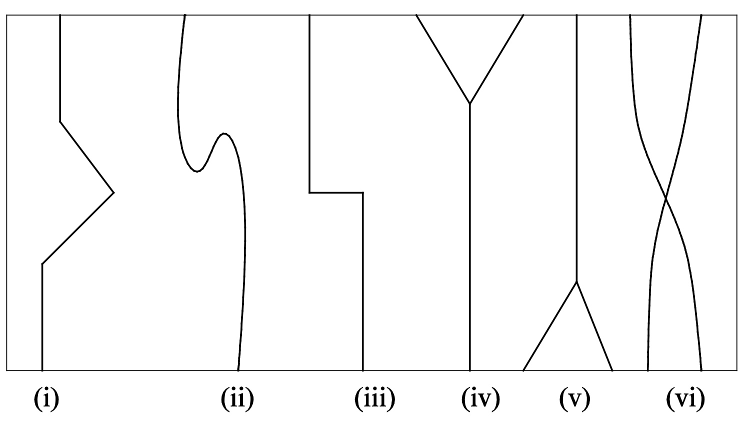
5. [4] Define a 3 x 3 filter that when convolved with a grayscale image will return a positive value if the average of the 4 nearest neighbors' intensity values is less than the center pixel's intensity value, and returns a negative or 0 value otherwise.

6. [13] The **seam carving** algorithm for image resizing iteratively removes the seam with the least energy.

(a) [3] How is the energy of a *seam* defined?

(b) [4] Let $E(I_t)$ be the average energy of all pixels computed from the reduced-size image I_t after t seams have been removed (*not* computed from the *original* image's energy minus all the seam pixels). Does the seam carving algorithm guarantee that $E(I_{t+1}) \geq E(I_t)$ for all iterations of seam carving? Briefly explain why or why not.

(c) [6] Which of the following are possible legal vertical seams found when using dynamic programming starting from the top row to the bottom row? Note: (iv) has two seams merging, (v) has two seams splitting, and (vi) has two seams crossing.



7. [7]

(a) [3] Which ***one*** of the following assumptions underlies creating an optically-correct **panorama** of a general 3D static scene from a set of input images using homographies (aka planar projective transformations) to align them?

(i) Rotation about the center of the camera's image sensor
(ii) Rotation about the center of the camera lens
(iii) Translation in the XY plane, assuming the Z axis points in the direction of the lens
(iv) Translation in the XZ plane, assuming the Z axis points in the direction of the lens

(b) [4] Instead of using blending to combine images into a panorama, an alternative approach would be to use optimal ***seam finding*** instead (as done in the overlapping region in texture synthesis) so that only a single source image is used to determine each output pixel's value. What is a possible noticeable artifact that might be visible in the panorama when using this approach, and what characteristic in the images will make this artifact more likely to occur?

8. [6]

(a) [3] A bear image was pasted into a swimming pool image in the left image below. A blending algorithm was applied to produce the result image on the right. Which **blending algorithm** was likely used: ***Laplacian pyramid blending*** or ***Poisson blending***? Briefly explain why.


(b) [3] What kind of visible artifacts would occur if the blending method used was ***feathering*** to combine the bear image and the swimming pool image above?

9. [9] Say we want to **warp** an image, *I*, into a new one, *J*, using the following homography (i.e., planar projective transformation):

$$\begin{bmatrix} sx \\ sy \\ s \end{bmatrix} = \begin{bmatrix} 2 & -1 & 8 \\ 1 & 2 & 10 \\ 1 & 0 & 4 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

(a) [3] If a point in one image is given by (*u*, *v*) = (2, 2), what are the *homogeneous* coordinates of the corresponding point in the other image?

(b) [3] What are the (real-valued) *Cartesian* coordinates of the corresponding point in the other image?

(c) [3] In order to prevent mapping multiple pixels in image *I* into a single pixel in image *J*, should (*u*, *v*) in the above equation be a pixel in image *I* or in image *J*?

10. [6] When creating panoramas for scenes that are far away from the camera's position, a simplification is to assume the images can be aligned by performing a **2D affine transformation**, i.e., a linear transformation from the coordinates of points in one image to points in a second image.

(a) [3] What is the form of the 3 x 3 matrix, H, in this case of a 2D affine transformation?

(b) [3] How many corresponding points between the two images are needed to estimate H?

11. [14] **Feature Point Detection and Description**

(a) [8]

(i) [4] How does the **SIFT feature point detector** compute a feature value at each position and scale? That is, describe what operation is used to compute each point's value.

(ii) [4] How does the **SIFT feature point detector** produce a locally sparse set of feature points, i.e., multiple points that are close together in the image are *not* all detected as feature points, from the values computed by the operator in (i)? Include in your answer which points are compared to a given feature point.

(b) [6] Yes or No: Is the SIFT feature point **descriptor** invariant to:

(i) [2] Arbitrary 2D image rotation?

(ii) [2] Arbitrary viewpoint change?

(iii) [2] Arbitrary camera focal length change?

12. [6] Given a set of n points specified by their 2D coordinates in an image, we'd like to automatically determine which *subset* of these points are approximately ***collinear***, i.e., can be approximated by a straight line. Recall that a line can be parameterized by two parameters, a and b, such that $y = ax + b$. Describe the main steps for how the **RANSAC** algorithm could be used to *find the line best fitting the inlier data points* when it is known that there are some "outlier" noise points, i.e., are not part of the line because they are greater than distance d from the line, but it is not known how many or which ones they are.