CS376: Computer Vision: MidTerm

1. The exam is 75 minutes.
2. You are allowed one page of hand written notes. No calculators, cell phones, or any kind of internet connections are allowed.
3. The sample mid-term is not representative of the true length or the point break-down of the final mid-term. It is intended to provide you an idea of the range of topics and the format of the mid-term.

1 Multiple Choice

Each question has ONLY ONE CORRECT OPTION and is worth 2 points. Please draw a circle around the option to indicate your answer.
1. In Canny edge detection, we will get more discontinuous edges if we make the following change to the hysteresis thresholding:

- (a) increase the high threshold
- (b) decrease the high threshold
- (c) increase the low threshold
- (d) decrease the low threshold
- (e) decrease both thresholds

Solution: (c)
Mean-shift is a non-parametric clustering method. However, this is misleading because we still have to choose

- (a) the number of clusters
- (b) the size of each cluster
- (c) the shape of each cluster
- (d) the window size
- (e) the number of outliers to allow

Solution: (d).
3.If you are unsure of how many clusters you have in your data, the best method to use to cluster your data would be

- (a) mean-shift
- (b) k-means
- (c) expectation-maximization
- (d) markov random field

- (e) none of the above are good methods

Solution: (a)

4. Normalized cuts is an NP-hard problem. To get around this problem, we do the following:

- (a) apply k-means as an initialization
- (b) allow continuous eigenvector solutions and discretize them
- (c) converting from a generalized eigenvalue problem to a standard one
- (d) constraining the number of cuts we make
- (e) forcing the affinities to be positive

Solution: (b)

5. To decrease the size of an input image with minimal content loss, we should

- (a) High-pass filter and down-sample the image
- (b) Crop the image
- (c) Apply a hough transform
- (d) Down-sample the image
- (e) Low-pass filter and down-sample the image

Solution: (e)

6. When applying a Hough transform, noise can be countered by

- (a) a finer discretization of the accumulator
- (b) increasing the threshold on the number of votes a valid model has to obtain
- (c) decreasing the threshold on the number of votes a valid model has to obtain
- (d) considering only a random subset of the points since these might be inliers

Solution: (b)

2 Long Answer

11.(9 points) Recursive Correlation

Recursive filtering techniques are often used to reduce the computational complexity of a repeated operation such as filtering. If an image filter is applied to each location in an image, a (horizontally) recursive formulation of the filtering operation expresses the result at location $(x+1, y)$ in terms of the previously computed result at location (x, y). A box convolution filter,B, which has coefficients equal to one inside a rectangular window, and zero elsewhere is given by:

$$B(x, y, w, h) = \sum_{i=0}^{w-1} \sum_{j=0}^{h-1} I(x+i, y+j).$$

where $I(x, y)$ is the pixel intensity of imageI at (x, y). We can speed up the computation of arbitrary sized box filters using recursion as described above. In this problem, you will derive the procedure to do this.

- The function J at location (x, y) is defined to be the sum of the pixel values above and to the left of (x, y), inclusive:

  $$J(x, y) = \sum_{i=0}^{x} \sum_{j=0}^{y} I(i, j).$$

  Formulate a recursion to compute $J(x, y)$. Assume that $I(x, y) = 0$ if $x < 0$ or $y < 0$. Hint: It may be useful to consider an intermediate image to simplify the recursion.

- Given $J(x, y)$ computed from an input image, the value of an arbitrary sized box filter (B_J) applied anywhere on the original image can be computed using four references to $J(x, y)$.

$$B_J(x, y, w, h) = a_J(?, ?) + b_J(?, ?) + c_J(?, ?) + d_J(?, ?)$$

Find the values of a, b, c, d and the ?'s to make this formula correct. Specifically, your answer should be the above equation with appropriate values for the above unknowns.

Solution:

- You can use the recursion:

$$J(x, y) = J(x - 1, y) + J(x, y - 1) - J(x - 1, y - 1) + I(x, y).$$

-

$$B_J(x, y, w, h) = J(x + w, y + h) - J(x - 1, y + h) - J(x + w, y - 1) + J(x - 1, y - 1).$$

12. (10 points) Linear Filter:

In this problem, you will explore how to separate a 2D filter kernel into two 1D filter kernels. Matrix K is a discrete, separable 2D filter kernel of size $k \times k$. Assume k is an odd number. After applying filter K on an imageI, we get a resulting imageI K.

- (1 point) Given an image point (x, y), find its value in the resulting image, $I_K(x, y)$. Express your answer in terms of I,k,K,x and y, You do not need to consider the case when (x, y) is near the image boundary.

  Solution:

$$I_K(x, y) = \sum_{i=1}^{K} \sum_{j=1}^{K} K_{ij} I(x - i + \frac{k}{2}, y - j + \frac{k}{2}).$$

- (5 points) One property of this separable kernel matrix K is that it can be expressed as the product of two vectors $g \in \mathbb{R}^{k \times 1}$ and $h \in \mathbb{R}^{1 \times k}$, which can also be regarded astwo 1D filter kernels. In other words, $K = gh$. The resulting image we get by first applying g and then applying h to the imageI is I_{gh}. Show that $I_K = I_{gh}$.

  Solution:

$$\begin{aligned}
I_K(x, y) &= \sum_{i=1}^{K} \sum_{j=1}^{K} K_{ij} I(x - i + \frac{k}{2}, y - j + \frac{k}{2}) \\
&= \sum_{i=1}^{K} \sum_{j=1}^{K} g_i h_j I(x - i + \frac{k}{2}, y - j + \frac{k}{2}) \\
&= \sum_{j=1}^{K} h_j \sum_{i=1}^{K} g_i I(x - i + \frac{k}{2}, y - j + \frac{k}{2}) \\
&= \sum_{j=1}^{K} h_j I_g(x, y - j + \frac{k}{2}) \\
&= I_{gh}(x, y).
\end{aligned}$$

- (4 points) Suppose the size of the image is $N \times N$, estimate the number of operations (an operation is an addition or multiplication of two numbers) saved if we apply the 1D filters g and h sequentially instead of applying the 2D filter K. Express your answer in terms of N and k. Ignore the image boundary cases so you do not need to do special calculations for the pixels near the image boundary.

Solution:

For the 2D filter, there are k^2 multiplication operations and $k^2 - 1$ addition operations for each pixel. In total, $N^2(2k^2-1)$. For each of the 1D filters, there are k multiplication operations and $k-1$ addition operations for each pixel. In total, $N^2(4k-2)$. So the number of operations saved is $N^2(2k^2-4k+1)$.

13.(10 points) Feature Detection and Description:

- (5 points) We want a method for corner detection for use with 3D images, i.e., there is an intensity value for each (x, y, z) voxel. Describe a generalization of the Harris corner detector by giving the main steps of an algorithm, including a test to decide when a voxel is a corner point.

  Solution:

  Similarly to the 2D case, we can estimate the average change inintensity for a shift (u, v, w) around a given voxel using a bilinear approximation given by

  $$E(u, v, w) = [u, v, w]M \begin{bmatrix} u \\ v \\ w \end{bmatrix}$$

  where M is a 3×3 matrix computed from partial derivatives in the 3 directions. Compute the 3 eigenvalues, λ_1, λ_2 and λ_3,of M, specifying an ellipsoid at the voxel that measures the variation in intensity in 3 orthogonal directions. Using the test in the Harris operator, mark the voxel as a corner point if $\lambda_1\lambda_2\lambda_3 - k(\lambda_1 + \lambda_2 + \lambda_3)$ is greater than a threshold.

- (5 points) The SIFT descriptor is a popular method for describing selected feature points based on local neighborhood properties so that they can be matched reliably across images. Assuming feature points have been previously detected using the SIFT feature detector, (i) briefly describe the main steps of creating the SIFT feature descriptor at a given feature point, and (ii) name three (3) scene or image changes that the SIFT descriptor is invariant to (i.e., relatively insensitive to).

  Solution:

  At each point where a SIFT "keypoint" is detected, the descriptor is constructed by computing a set of 16 orientation histograms based on 4×4 windows within a 16×16 pixel neighborhood centered around the keypoint. At each pixel in the neighborhood, the gradient direction (quantized to 8 directions) is computed using a Gaussian with σ equal to 0.5 times the scale of the keypoint. The orientation histograms are computed relative to the orientation at the keypoint, with values weighted by the gradient magnitude of each pixel in the window. This results in a vector of $128(= 16x8)$ feature values in the SIFT descriptor. (The values in the vector are also normalized to enhance invariance to illumination changes.) (ii) Because the SIFT descriptor is based on edge orientation histograms, which are robust to contrast and brightness changes and are detected at different scales, the descriptor is translation,rotation, scale, and illumination (both intensity change by adding a constant and intensity change by contrast stretching) invariant. It is not invariant to significant viewpoint changes.

14.(10 points) Hough Transform and RANSAC:

Assume we have a 3D point cloud produced by a single laser scanner. The point cloud contains a single dominant plane (e.g., the front wall of a building) at unknown orientation, plus smaller numbers of other scene points (e.g., from trees, poles and a street) that are not part of this plane. As you know, the plane equation is given by $ax + by + cz + d = 0$.

- Define a Hough transform based algorithm for detecting the orientation of the plane in the scene. That is, define the dimensions of your Hough space, a procedure for mapping the scene points (i.e., the (X, Y, Z) coordinates for each pixel) into this space, and how the plane's orientation is determined.

  Solution:

  Assuming the plane is not allowed to pass through the camera coordinate frame origin, we can divide by d, resulting in three parameters, $A = a/d$,$B = b/d$,and $C = c/d$ that define a plane. Therefore the Hough parameter space is three dimensional corresponding to possible values of A ,B,and$C$. Assuming

we can bound the range of possible values of these three parameters, we then take each pixel's (X, Y, Z) coordinates and increment all points $H(p, q, r)$ in Hough space that satisfy $pX + qY + rZ + 1 = 0$. The point (or small region) in H that has the maximum number of votes determines the desired scene plane.

- Describe how the RANSAC algorithm could be used to detect the orientation of the plane in the scene from the scene points.

  Solution:

  *S*tep 1: Randomly pick 3 pixels in the image and, using their (X, Y, Z) coordinates, compute the plane that is defined by these points.

  *S*tep 2: For each of the remaining pixels in the image, compute the distance from its (X, Y, Z) position to the computed plane and, if it is within a threshold distance, increment a counter of the number of points (the "inliers") that agree with the hypothesized plane.

  *S*tep 3: Repeat Steps 1 and 2 many times, and then select the triple of points that has the largest count associated with it.

  *S*tep 4: Using the triple of points selected in Step 3 plus all of the other inlier points which contributed to the count, recompute the best planar fit to all of these points.

3 Short Answer

15. **(5 points)** Describe two applications of non-maximum suppressing.
Solution:
Canny edge detector and Harris edge detector.

16. **(5 points)** You are using k-means clustering in color space/itensity to segment a natural image with diverse appearance. However, you notice that although pixels of similar color/intensity are indeed clustered together into the same clusters, there are many discontiguous regions because these pixels are often not directly next to each other. Describe a method to overcome this problem in the k-means framework.
Solution:
Concatenate the coordinates (x,y) with the color features as input to the k-means algorithm.

17. **(5 points)** Suppose you are using RANSAC to detect lines in an image. You have the choices of voting lines from two sample points and voting lines from one sample point with arbitrary orientation. Please briefly discuss the tradeoffs.
Solution:
Using two sample points leads to more discriminative results but requires more votes. Using one point + normal is more efficient but normal computation can be error-prune. Using one point + arbitrary orientation is more efficient but the voting results are less discriminative.

18. **(5 points)** In mean-shift clustering, what is the relation between σ that defines the density functions and the number of resulting clusters.
Solution:
Increase σ will reduce the number of resulting clusters.

19. **(6 points) True / False : Fundamental matrix estimation**

- (2 points) What is the rank of the fundamental matrix?

  Solution:

  The rank is 2.

- (4 points) In the 8-point algorithm, what math technique is used to enforce the estimated fundamental matrix to have the proper rank? Explain how this math technique is used to enforce the proper matrix rank.

  Solution:

  In the 8-point algorithm, SVD can be used to enforce the estimated F has rank 2. Specifically, we compute the SVD decomposition $F = U\Sigma V$, and we then zero out diagnals of Σ except for the two largest singular values to obtain Σ. We can reconstruct $F = U\Sigma V$.

20. **(3 points)** True / False : Harris Corner Detector is rotation invariant. Why or why not?
Solution: True.

Rotating the image essentially applies a unitary transformation to the M matrix for each pixel, and this transformation does not really change the eigenvalues of M. So it does not change the detection results.

21. **(3 points)** True / False: Harris Corner Detector is scale invariant. Why or why not?
Solution: False.

The M matrix is dependent on the window size. Choosing different window size would significantly impact the cornerness score.

22. **(3 points)** True/ False: SIFT descriptor is both rotation invariant and scale invariant. Why or why not?
Solution: True.

SIFT descriptor is rotation invariant because the region is oriented with respect to the gradient direction, and the scale is automatically determined by detecting the critical point in scale at each pixel.

23. **(3 points)** True / False : If we initialize the k-means clustering algorithm with the same number of clusters but different starting positions for the centers, the algorithm will always converge to the same solution. Why or why not?
Solution: False.

Different initializations will result in different clusters because they are local minima.