Exam Handed out: 2:00 p.m. Monday in Room 1112 A.V. Williams
Due back: 3:00 p.m. Wednesday in Room 1112 A.V. Williams

Instructions:
This exam will count for 25 % of the total grade.

Solutions will be posted online on Wednesday. We hope to have the grading done by Friday.

You may use any books/material you want. Please do not take help of any one, or collaborate.

Please explain all steps clearly.

Please do not post questions to the class mailing list on any of the questions. If you think there is an issue with any of the questions, address them to either Dr. DeMenthon or Duraiswami. (daniel@cfar or ramani@umiacs)

Good Luck!

Questions

1. Suppose we have a camera that is undergoing a purely translational motion, and that the camera, with a focal length of 100, is moving with velocity
$$(U,V,W) = (20, 30, 50).$$
a) Generally, what is the focus of expansion for a moving camera? [0.5 pt]
b) What is the location of the focus of expansion for this particular motion? [1 pt]
c) What is the pattern of the optical flow vectors for this type of motion? [0.5 pt]
d) What is the aperture problem in computing optical flow? How is it related to the linear constraint on optical flow ("brightness constraint") determined by the differential technique for computing flow? [1 pt]

2. Contours of maximum rate of change in an image can be found by locating directional maxima in the gradient magnitude of the image or (almost) equivalently, by finding zero-crossings in the Laplacian of the image.
(a) Why is it considered important to convolve the image with a Gaussian before computing the Laplacian? [0.5pt]
(b) Because convolution and the Laplacian operator are both linear, convolving the image with a Gaussian and then taking the Laplacian, $(\nabla^2(G*I))$, is exactly equivalent to taking the Laplacian of the same Gaussian and then convolving that with the image, $(\nabla^2 G)*I$. Why does the second formulation lead to a more efficient implementation? [1 pt]
(c) The Laplacian of a Gaussian can be approximated by the difference of two Gaussian kernels with appropriately chosen standard deviations. As a result, show that the computation can be implemented as $(\nabla^2 G)*I \approx G_1*I - G_2*I$ [1pt]

(d) Why is the difference of Gaussians implementation even more efficient than the $(\nabla^2 G)*I$ Laplacian of Gaussian implementation? [0.5 pt]

3. Suppose that a camera is at an initial position in which its coordinate system coincides with the coordinate system of the scene, so that the camera matrix can be expressed as **P** = **K** [**I** | **0**], where **K** is the calibration matrix. The camera takes a view of the scene, then undergoes a pure translation **T**, and then takes a second view. There is no change in the calibration between the two views and there is no camera rotation during the displacement.
(a) Assuming that **T** is expressed in the coordinate system of that camera, what is the form of the camera matrix after translation, in relation to **K** and **T**? [1 pt]
(b) Show that the fundamental matrix for the two images depends only on the epipole **e'** of the second camera. [1 pt]
(c) An image point **x** in the initial view moves to a point **x'** because of the translation. The point **x'** is on the epipolar line of **x**, which passes through the epipole **e'**. Show that the point **x** also belongs to that epipolar line, and therefore that for pure translation displacements, points move along epipolar lines. (The epipole in the case of pure translation does not move from view to view and corresponds to the focus of expansion of optical flow). [1 pt]
(d) The translation is now assumed parallel to the x-axis of the coordinate system of the camera in its initial position. What form does the fundamental matrix take in this special case? [1 pt]

4. A calibrated camera with normalized image coordinates has a camera center **C** and a camera matrix equal to **P = [I | 0]**. Its coordinate system coincides with the coordinate system of the scene. For concreteness, the direction of the columns of its CCD array (the y axis of the scene coordinate system) is vertical. This camera moves to a second position by a pure planar motion in a horizontal plane. Its camera center occupies a new position **C'**. The horizontal translation expressed in the scene coordinate system is **T** = [T_x, 0, T_z] and the rotation is characterized by a single angle θ around the vertical direction. The notations s and c are used for the sine and the cosine of θ.
(a) Express the coordinates of the initial camera center **C** in the new camera coordinate system of the camera after the planar displacement. Use these coordinates to express the camera matrix **P'** of the camera after the displacement as a function of T_x, T_z, s and c. [1 pt]
(b) Write the two quadratic conditions that must be met simultaneously by a scene point **X** with components (X, Y, Z) in order for the scene point to have image points at the same position in the two images (i.e., so that the disparity of the two image points is zero) (Hint: write an equality between the *inhomogeneous* coordinates of the two image points.) [1 pt]
(c) Show that the first condition corresponds to a vertical cylinder of revolution and the second condition corresponds to a set of two planes. Show that the simultaneous satisfaction of these two conditions implies that the locus of scene points with zero disparity consists of a circle and a vertical line. [1 pt]

(d) Show that the circle passes through the two camera centers and the intersection of the two principal axes. [1.5 pt]
(e) What is the intersection of the vertical line with the plane of the circle? [1.5 pt]

5. Consider the Matlab function for computing feature points available at **`http://www.umiacs.umd.edu/~ramani/cmsc828d/harris.m`**
(a) Describe step-by-step the operations performed by this function on an image and the reasons why these operations are performed. [2 pts]
(b) Consider the first and last images of the Yosemite sequence (number 2 and 16) (**`http://www.umiacs.umd.edu/~ramani/cmsc828d/yos.02.tif`** & **`http://www.umiacs.umd.edu/~ramani/cmsc828d/yos.16.tif`**).
Automatically detect feature points in each image using Matlab and the **`harris.m`** function. Write Matlab code that sorts these points, selects 20 best feature points, matches corresponding points between the two images using some proximity and similarity criteria, and draws vectors joining them. Print your results. Here there is no right or wrong answer. However, only the most sensible and creative approaches will obtain the maximum grade. [3 pts]

6. Recall the definition of an eigenvalue and an eigenvector $\mathbf{Ax}=\lambda\mathbf{x}$.
We can define $$f(\mathbf{x})=\frac{\mathbf{x}^t\mathbf{Ax}}{\mathbf{x}^t\mathbf{x}}.$$
Given a matrix **A**, we can view $f(\mathbf{x})$ as a scalar valued function of **x**.(This function is called the Rayleigh quotient in the literature.)
a) Show that stationary points of the $f(\mathbf{x})$ with respect to **x** are eigenvectors of **A**, and that the corresponding values of $f(\mathbf{x})$ are eigenvalues.
(Hint: Recall the definition of a stationary point, and remember that we are looking for a nonzero **x**.) [1.5 pt]
b) Write a function that returns the Rayleigh quotient of a 2 by 2 matrix **A**, given a vector **x**.
Test your code using the square matrix $$\mathbf{A}=\begin{bmatrix} 3 & \sqrt{3} \\ \sqrt{3} & 3 \end{bmatrix}.$$
Use the Matlab **`mesh`** function to plot the negative of the Rayleigh quotient, $-f(\mathbf{x})$, where **x** is the 2D vector $[x_1, x_2]'$ with component values in the range $-1.5 \le x_1, x_2 \le 1.5$ (You may want to avoid the region around (0,0) by choosing your points judiciously). [1 pt]
c) Find the **maximum** of this function using the Matlab function **`fminsearch`.** Compare the values you obtain for the eigenvector and eigenvalue using the Matlab function **`eig`**. Are the values different? Why? [1.5 pt]