

HCI/ComS 575X: Computational Perception

Homework 1, Part I (Basic Linear Algebra)

Out: Wed. Jan 18, 2006

Due: Mon. Jan 30, 2006

For extra credit submit *****part I only***** by Fri. Jan 20 before 4pm.

(submit it to Karen in VRAC front office.)

Note: Not all homeworks are created equal :(

These problems are designed to test your background knowledge in linear algebra and statistics.

Please show the intermediary steps of your calculations (not just the final answers).

1. Calculate the following determinants.

$$(a) \begin{vmatrix} \sqrt{2} & \sqrt{6} \\ 4 & \sqrt{3} \end{vmatrix} \quad (b) \begin{vmatrix} a-3 & 5 \\ -3 & a-2 \end{vmatrix} \quad (c) \begin{vmatrix} -1 & 1 & 2 \\ 3 & 0 & -5 \\ 1 & 7 & 2 \end{vmatrix} \quad (d) \begin{vmatrix} c & -4 & 3 \\ 2 & 1 & c^2 \\ 4 & c-1 & 2 \end{vmatrix}$$

2. Show that the value of the determinant does not depend on θ .

$$\begin{vmatrix} \sin\theta & \cos\theta & 0 \\ -\cos\theta & \sin\theta & 0 \\ \sin\theta - \cos\theta & \sin\theta + \cos\theta & 1 \end{vmatrix}$$

3. Calculate the inverse of the given matrix.

$$(a) \begin{bmatrix} 1 & 4 \\ 2 & 7 \end{bmatrix} \quad (b) \begin{bmatrix} -3 & 6 \\ 4 & 5 \end{bmatrix} \quad (c) \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix} \quad (d) \begin{bmatrix} k_1 & 0 & 0 & 0 \\ 0 & k_2 & 0 & 0 \\ 0 & 0 & k_3 & 0 \\ 0 & 0 & 0 & k_4 \end{bmatrix}$$

4. Solve the following systems of equations.

$$(a) \begin{cases} x_1 + x_2 = 2 \\ 5x_1 + 6x_2 = 9 \end{cases} \quad (b) \begin{cases} x_1 + 3x_2 + x_3 = 4 \\ 2x_1 + 2x_2 + x_3 = -1 \\ 2x_1 + 3x_2 + x_3 = 3 \end{cases}$$

5. Let $u = (3, 2, -1)$, $v = (0, 2, -3)$, and $w = (2, 6, 7)$. Compute:

$$(a) u \cdot v \quad (b) \|u\| \quad (c) v \times w \quad (d) u \times (v \times w) \quad (e) (u \times v) \times w$$

6. Find the orthogonal projection of u on a (i.e., find $proj_a u = \frac{u \cdot a}{\|a\|^2} a$)

(a) $u = (6, 2), a = (3, -9)$

(b) $u = (3, 1, -7), a = (1, 0, 5)$

7. Find the angle θ between the two vectors u and v .

(a) $u = (2, 3), v = (5, -7)$

(b) $u = (1, 0, 0), v = (1, 1, 1)$

8. Find an equation for the plane passing through the given points.

(a) $P(-4, -1, -1), Q(-2, 0, 1), R(-1, -2, -3)$ (b) $P(5, 4, 3), Q(4, 3, 1), R(1, 5, 4)$

9. Find all values of λ for which the determinant of the matrix is equal to 0.

(a) $\begin{vmatrix} \lambda - 2 & 1 \\ -5 & \lambda + 4 \end{vmatrix}$ (b) $\begin{vmatrix} \lambda - 4 & 0 & 0 \\ 0 & \lambda & 2 \\ 0 & 3 & \lambda - 1 \end{vmatrix}$

10. Find the eigenvalues of the following matrices.

(a) $\begin{bmatrix} 4 & 0 & 1 \\ -2 & 1 & 0 \\ -2 & 0 & 1 \end{bmatrix}$ (b) $\begin{bmatrix} 3 & 0 & -5 \\ \frac{1}{5} & -1 & 0 \\ 1 & 1 & -2 \end{bmatrix}$

11. Given the vector $v = [x, y]^T$ find a rotation matrix R which rotates the vector by 60° counter-clockwise. Give the values for the vector $v' = [x', y']^T$ in terms of x and y , where $v' = Rv$.

12. Write down the analytical form of the 3D rotation matrices R_x , R_y , and R_z which rotate a vector about the X , Y , and Z axis, respectively.

13. By example, show that the 3D rotation matrices are not commutative.

HCI/ComS 575X: Computational Perception

Homework 1, Part II (Basic Matlab)

Out: Wed. Jan 18, 2006

Due: Mon. Jan 30, 2006

Submit this part in electronic form on your section of the class wiki.

This part requires you to use matlab to perform some simple imaging and plotting operations. Please refer to the matlab tutorial from last week (or google) if you have any problems.

1. Edge Detection

Find a JPEG image. Load it in matlab. Perform edge detection on it using your favorite edge detection operator. Save the resulting edges image. Post your original image, your edges image, and your matlab code on your wiki page.

2. Try Some Other Functions

Pick your favorite function from matlab's image processing toolkit. Perform the same procedure as in 1 but replace the edge detection function with your new function. Now choose two more functions and do the same. Post your original image, your three resulting images, and your code on the wiki.

3. Plotting a Gaussian

Write a short matlab program that plots the probability density function of a 1D Gaussian function with mean $\mu = 5$ and standard deviation $\sigma = 2$.

Write another matlab program that plots the probability density function of a 2D Gaussian function with mean $\mu = (\mu_x, \mu_y) = (0, 0)$ and a diagonal covariance matrix Σ as shown below:

$$\Sigma = \begin{bmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{bmatrix}$$

where $\sigma_x^2 = 1$ and $\sigma_y^2 = 2$. Hint: In this case the result should be a surface.

When you are done post your two programs and your resulting plots (matlab can save them for you; check the figure window menu) on the wiki.

HCI/ComS 575X: Computational Perception

Homework 1, Part III (Basic OpenCV)

Out: Wed. Jan 18, 2006

Due: Mon. Jan 30, 2006

Submit this part in electronic form (see details below).

To introduce you to OpenCV, we will do three fairly simple tasks. For this part you must submit your results in electronic form on your section of the class Wiki. The TA has prepared some templates that you can use to simplify your task (<http://www.hci.iastate.edu/575x/doku.php?id=tylerstreeter:homework:hw1>)

1. Edge Detection

Start with finding a JPEG image of yourself. If you don't have one already find a webcam and take one. Now write a complete C/C++ code using openCV that reads your image, performs edge detection on it, and then saves the result to another JPEG file. Post your original image, your edges image, and your code on your section of the wiki.

2. Putting a caption on a movie

For this task you must first find a digital movie (on Windows you can only use the AVI format; on Linux you can use MPEG if you compiled openCV with ffmpeg; see the online tutorials for more details). If you don't have a movie we posted some that you can download from the wiki page. Once you have the movie, write a program that uses openCV and reads the video one frame at a time, prints your name on each frame, and saves the video to another file with your name on each frame.

Post your original video, your results video, and your code on your wiki page. If you choose to use your own video please choose a small one ($\leq 2M$) to save web space. If all else fails, just compress your videos in zip or tar.gz files before you upload them.

3. Blurry Videos

This task is very similar to the previous one. Use the same input video as before. This time, however, you need to add a slider to your openCV window which controls the amount of blurring/smoothing performed on each frame. Your application should be able to save the movie while you actively change the amount of blurring by dragging the slider with the mouse. The resulting movie should start normal, go blurry, and then go back to normal.

Post only the resulting blurry movie on your wiki page (the original should be the same as in 2). Also, post your code.

That's it. Good Luck!