UNIVERSITY OF TEXAS AT ARLINGTON
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

6367 **COMPUTER VISION** **SPRING 2020**

ASSIGNMENT 1 (100 POINTS)
ASSIGNED: 1/28/2020 DUE: 2/11/2020

This assignment constitutes 10% of the course grade. You must work on it individually and are required to submit a PDF report along with the MATLAB scripts described below.

Problem 1 (10 points)

(a) (5 points) Prove that if A is an $n \times n$ (real) symmetric matrix, then there exists an $n \times n$ (real) orthogonal matrix U and $n \times n$ (real) diagonal matrix D such that $A = U \cdot D \cdot U^T$.

(b) (5 points) Given

$$\begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix},$$

plot the regions

$$\{Ax \mid x \in \mathbb{R}^2 \text{ and } ||x||_2 = 1\},$$

and

$$\{Ax \mid x \in \mathbb{R}^2 \text{ and } ||x||_2 \leq 1\}.$$

Explain your solution.

Problem 2 (10 points)

Show that the determinant of a rotation matrix is ± 1.

Problem 3 (10 points)

(a) (5 points) Let R_1 and R_2 be two rotation matrices on the plane. Prove or disprove the following: $R_1 R_2 = R_2 R_1$.

(b) (5 points) Let R_1 and R_2 be two rotation matrices in 3D space. Prove or disprove the following: $R_1 R_2 = R_2 R_1$.

Problem 4 (10 points)

Let R be a 3D rotation matrix. Claim: 1 is an eigenvalue of R. Is the claim true? If so, what is the physical meaning of the corresponding eigenvector?

Problem 5 (10 points)

Write a MATLAB function `[A, theta] = get_axisangle(R)` that takes a rotation matrix `R` as input and returns the rotation axis `A` and angle `theta`.

Submission Instructions: *For Problems 1, 2, 3, and 4, submit the typed solutions in the report. For Problem 5, submit a MATLAB function get_axisangle.m that performs the necessary operations.*

Problem 6 (50 points)

The objective of this problem is to introduce basic image processing operations in MATLAB. Load the accompanying "board.tif" image in MATLAB.

(a) (5 points) Extract the rectangular block of the image between $(200, 90)$ and $(300, 180)$ corresponding to the crystal in the image. Display this block in a separate figure.

Solve **(b)** and **(c)** in the following three different ways: (i) using nested for loops and accessing each pixel individually, (ii) using MATLAB matrix operations, (iii) using built-in image processing functions `rgb2gray` and `im2bw`. Use the MATLAB profiler to compare the running times.

(b) (10 points) Convert the image from RGB to grayscale. For each pixel, take the average of the R, G, and B values as the grayscale value.

(c) (10 points) Convert the grayscale image to a binary image using the mean grayscale value as the threshold. Display both the grayscale and binary images in the same window.

(d) (25 points) Smooth the grayscale image created above using a 7×7 averaging filter. This means that for each pixel at location (i, j), place a 7×7 window centered at (i, j) and replace the value of the pixel with the average of the values of the pixels in the window. Decide how you will handle pixels close to the image boundary. Solve this problem in the following two different ways: (i) using for loops, (ii) using the MATLAB function `conv2`. Display the smoothed images.

Submission Instructions: *Submit a single MATLAB file, problem_2.m, that reads in the image and performs all of the operations stated above. Please take care to generate all the figures in new windows. You are also required to embedded all of the images in the report (do not submit the images separately) along with a comparison of the runtimes. The MATLAB command* `print` *may be helpful in this regard. Each sub-problem must be marked separately and clearly.*

Extra Credit (20 points)

(a) (10 points) We denote by $\sigma_i(B)$ the ith singular value of B (sorted in descending order). Prove that if A_1 and A_2 are $m \times n$ matrices, then for all i and j in $\mathbb{N}$:

$$\sigma_{i+j-1}(A_1 + A_2) \leq \sigma_i(A_1) + \sigma_j(A_2).$$

(b) (10 points) Choose an image of size at least 512×512. Apply SVD compression to the image with 3, 10, 20 and 40 top singular values and vectors. Plot the original image and the "compressed" ones. In addition, make a table with relative errors and compression ratios (i.e., ratios between sizes of the compressed SVD nonzero components to the sizes of the original matrices) for each of the images.

Submission Instructions: *For (a), submit the typed solution in the report. For (b), embed the plots and table in the report (do not submit the images separately). Please take care to generate all the figures in new windows. The MATLAB command* `print` *may be helpful in this regard.*