CSC 418/2504 Computer Graphics, Fall 2017
Assignment 2 (15% of course grade)

Due 11:59pm on Wed., Nov. 1, 2017.

Part A [50 marks in total]

Below are 4 exercises covering different topics from the weeks of class upto reading week. They require thought, so you are advised to consult the relevant sections of the textbook, the online lecture notes and slides, and your notes from class well in advance of the due date. Your proofs and derivations should be clearly written, mathematically correct, and concise.
All questions require showing the steps toward the solution, and marks will be subtracted if this is not the case. Even if you cannot answer a question completely, it is very important that you show your (partial) answers and your reasoning. Otherwise your TA will ***not*** *be able to award you partial marks.*
Both Part A and Part B must be done individually and electronically submitted. Part A must be in PDF format, by scanning your handwritten solution or by using LaTeX/word to typeset it.

1. *Transformations* [15 marks]

   [9 marks] A 2D Affine transformation, is completely specified by its effect on three non-colinear points, i.e., by how it maps a triangle into another triangle. Find the 2D affine transformation that maps points $\mathbf{a_1}$, $\mathbf{a_2}$, and $\mathbf{a_3}$ into points $\mathbf{b_1}$, $\mathbf{b_2}$, and $\mathbf{b_3}$ respectively.
   Under what conditions (for the points), is this mapping fully determined?

   [2 marks] How many point mappings need to be specified to completely determine a general 2D Homography? A 2D similarity transform?

   [4 marks] Are the centroid (average of the three points) and ortho-center (intersection point of the altitudes) of a triangle affine invariant? Prove or provide a counterexample.

2. *Viewing and Projection* [10 marks]

   [3 marks] Why is the image formed in a pin hole camera inverted? (no more than a few sentences)

   [3 marks] Given a 3D camera position $\mathbf{c}$, a point along the viewing direction at the centre of the screen $\mathbf{p}$, and a vector parallel to the vertical axis of the screen $\mathbf{u}$, compute the world to camera transformation matrix.

   Given a 3D camera at the origin viewing along the $\mathbf{z}$-axis, and a projection plane (screen) located at distance d,

      [2 marks] what is the 2D (x,y) point that is the perspective projection of a 3D point $\mathbf{p} = (p_x, p_y, p_z)$?

      [2 marks] Under what conditions will a family of lines parallel to the vector $\mathbf{v} = (v_x, v_y, v_z)$ remain parallel after this perspective projection?

[2 marks] When this condition is not met, do all lines converge at a single 2D point? If so, which point? If not, provide counterexample?

3. *Surfaces* [12 marks]

The tangent plane of a surface at a point is defined so that it contains all tangent vectors. In this exercise, you will verify that a specific tangent vector is contained in the tangent plane. Let the surface be a torus in 3D (Figure 1) defined by the implicit equation:

$f(x, y, z) = (R - sqrt(x^2 + y^2))^2 + z^2 - r^2 = 0$, where $R > r$.

[3 marks] Give a surface normal at point $\mathbf{p} = (x, y, z)$, using the surface implicit equation.
[3 marks] Give an implicit equation for the tangent plane at $\mathbf{p}$.
[3 marks] Show that the parametric curve $\mathbf{q}(\lambda) = (R \cos \lambda, R \sin \lambda, r)$ lies on the surface.
[3 marks] Find a tangent vector of $\mathbf{q}(\lambda)$ as a function of λ.
[3 marks] Show this tangent vector at $\mathbf{q}(\lambda)$ to lie on the implicit equation of the tangent plane.

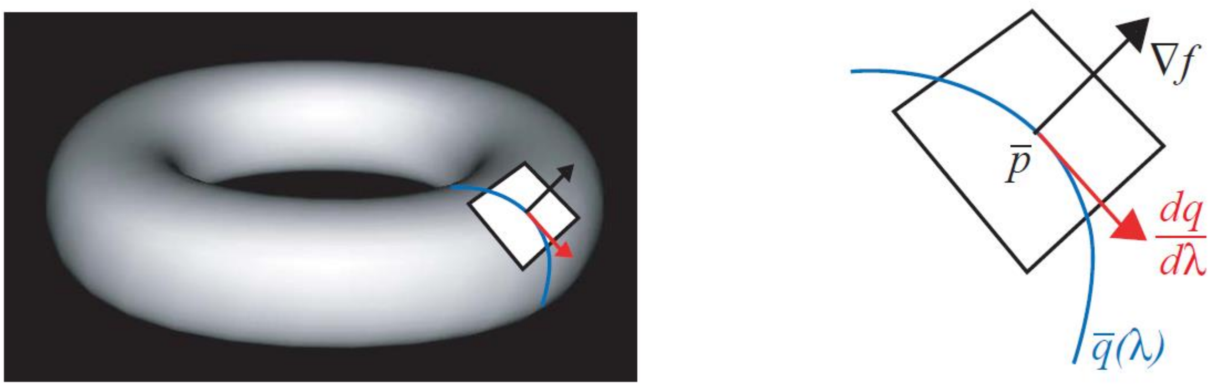

Figure 1: *Left:* Torus, showing a tangent plane, normal and 3D curve at a point. *Right:* Close-up.

4. *Visibility* [10 marks]

Figure 2 below shows a 2D “scene” composed of seven oak trees. A camera “eye” with viewing frustum gazes into the scene.

[2 marks] Does a hierarchical axis-aligned bounding box (AABB) tree of the seven oaks depend on the location and orientation of the viewing frustum?

[4 marks] Partition the oaks into a balanced binary tree of AABBs. Draw the each bounding box ***and*** draw a schematic of the tree using the letters on each oak. Leaf nodes should contain a single oak. At each node of the tree, sort and split the contained oaks along the longest bounding box direction.

[4 marks] Assuming a fast `box_frustum_intersection` subroutine, write a pseudocode algorithm to traverse a given AABB tree and gather all oaks that may intersect with the frustum.
Hint: Using your AABB tree for the example below, only E should be returned.

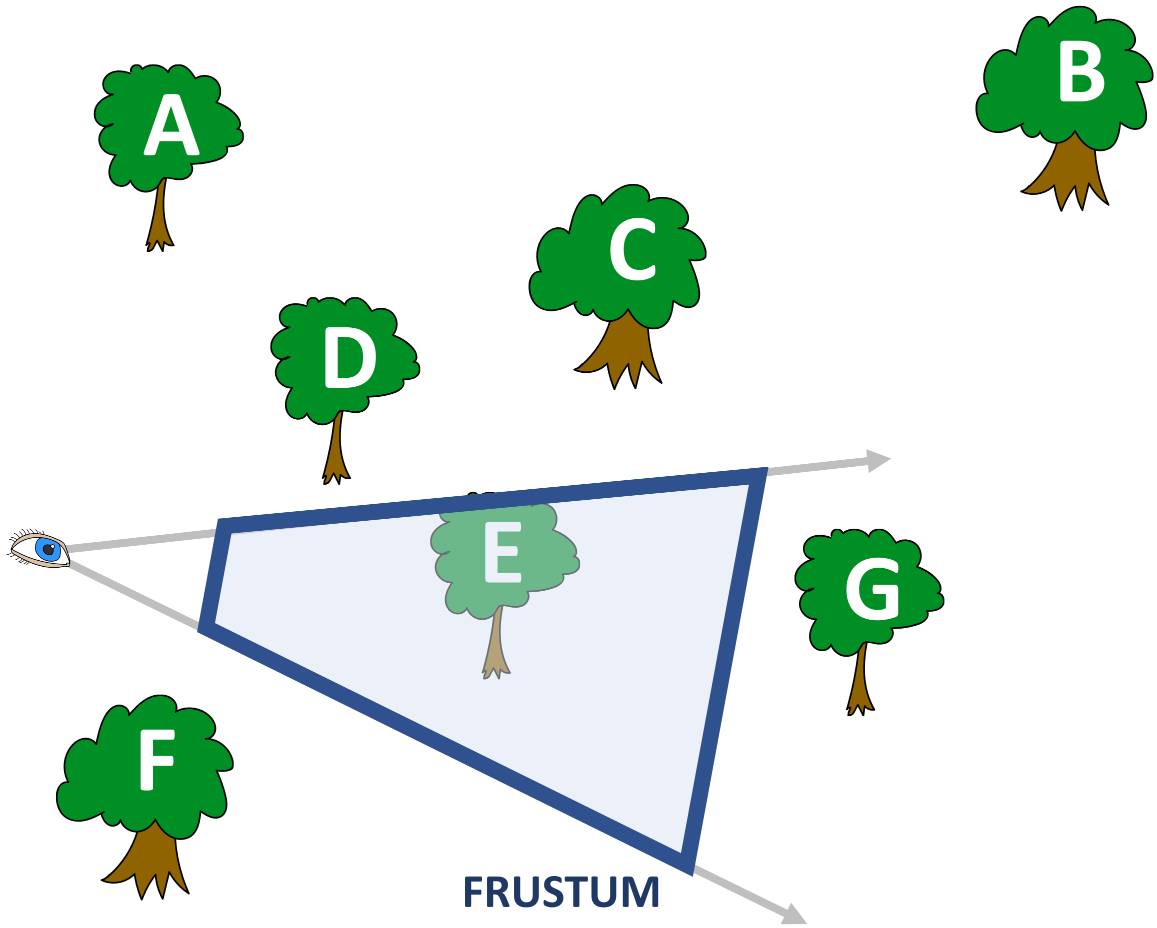

Figure 2: A 2D scene composed of seven oak trees viewed within a frustum (blue parallelogram).