CMSC425 Fall 2019
Unit: Geometry and Geometric Programming
Objectives

Here's what you should know from the "Geometry and Geometric Programming" part of the course. We will add applications of this material through the semester.

Readings on the web page:

Mount Lecture 4, Geometry and Geometric Programming
Mount Lecture 5, More on Geometry and Geometric Programming
Handouts on: Lines/Planes notes, Tweening notes

Background:

We do assume you can do the following linear algebra, so look it up if it's been a while.

Multiply two matrices, typically of dimensions 2x2 to 4x4.
Compute determinant of 2x2 and 3x3 matrices (for now).
Transpose a matrix or vector.
Figure out if a set of vectors are linearly independent (or dependent)

Most of our linear algebra will be in 2 and 3 dimensions, with excursions to 4d, with the potential to go to matrices for n data points.

Support:

We recommend the following software:

Latex for find
Matlab or Octave for performing linear algebra calculations
Octave online (https://octave-online.net) is easy to access for quick calculations

Objectives:

Here's a specific list of concepts and problems you should know after this unit:

I. Basic Affine and Euclidean objects and operations

- A. Affine geometry
  - i. Basic objects as scalar, points and free vectors
  - ii. Affine operations as on page 3 of Mount lecture 4
  - iii. Why we don't strictly add or subtract points
  - iv. Carry out vector additions and subtractions by diagram
- B. Affine and convex combinations and their applications
  - i. Affine - sum of points with coefficients sum to 1
  - ii. Convex - sum with coefficients sum to 1, coefficients all >= 0
  - iii. Applications: midpoint of line, center of triangle, etc
  - iv. Convex gives all points *inside* shape (eg, line, triangle)
  - v. Affine gives all points on a line or plane defined by points

- C. Euclidean operations
    - i. Perp vector (v = <x,y> => v_perp = <-y,x>)
    - ii. Dot product
        - a. Properties: positiveness, symmetry, bilinearity
        - b. Use to compute magnitude of vector
            - i. for normalization of vector
            - ii. for distance between two points
        - d. Use cosine formula to
            - i. Compute angle between two vectors
            - ii. Compute cos of that angle directly
        - e. Use it to determine sign of angle between two vectors
            - i. Orthogonal if u dot v = 0
            - ii. Acute (<90) if u dot v > 0
            - iii. Obtuse (>90) if u dot v < 0
        - f. Orthogonal projection
            - i. Compute with perp vector
            - ii. Compute with dot product as in Mount notes
    - iii. Cross product
        - a. Interpret using right/left handed rule
        - b. Compute using determinant of matrix
        - c. It is skew symmetric and non-associative
        - d. u x v is perpendicular to u and v
        - e. Sin rule for cross product
        - f. Sin rule => u || v means u x v = 0
        - g. If |u|=|v|=1 then |u x v| = 1
- D. Coordinate frames and homogeneous coordinates
    - i. Coordinate frame as origin plus linearly ind. vectors
    - ii. Representation of point as (x,y,w) with w = 1
    - iii. Representation of vector as (x,y,w) with w = 0
    - iv. These representations preserved by affine operations
        - a. v = p - p = (x,y,0)
        - b. 2*v = (x,y,0)
        - c. v-v or v+v = (x,y,0)
        - d. NOT by addition of points p+p = (x,y,2)
    - v. Preserved by convex and affine combination
        - a*p1 + b*p2 = (x,y,1) iff a+b = 1

II. Computation and representation of geometric objects

- A. Representation of curves
    - i. Implicit ($x^2 + y^2 = R^2$)
    - ii. Explicit ($y = mx + b$)
    - iii. Parametric ($P = P_0 + t*v$)
    - iii. Describe advantages of parametric questions

- B. Representing lines, rays and line segments
    - i. P = P0 + v * t (t for time) (vector parametric)
    - ii. If t in [0,1], line segment
    - iii. If t in [0,INF], ray
    - iv. If t in [-INF,INF], line
    - v. P = t*P0 + (1-t)*P1
    - vi. Convert between vector, blending and implicit version
    - vii. Point normal form (hint, use perp vector)
- C. Planes (see handout "Notes on line and plane representations)
    - i. Blending parametric representation from three points
    - ii. Vector parametric representation from point + 2 vectors
    - iii. Implicit equation with normal
    - iv. Using cross product to find normal to triangle/polygon
- D. Circles (done in class)
    - i. px = R*cos(t) + cx, py = R*sin(t) + cy, t in range [0,2π]

III. Applications:

- A. Basic applications
    - i. Represent shapes
    - ii. Represent kinematic motion
    - iii. Calculate force vectors for physics motion
    - iv. Represent light as rays
    - v. Calculate collision/distance between shapes
    - vi. Place objects in space by equation
- B. Specific applications
    - i. Distance of point to line
    - ii. Distance of point to plane
    - iii. Find perpendicular bisector
    - iv. Find point of intersection of two lines
    - v. In 2D intersect line (or line seg or ray) with plane (or triangle or circle)
    - vi. Find simplicity of polygon (show edges non-intersecting)
    - vii. Find winding direction of polygon
    - iix. Find convexity (concave, convex) of polygon
    - ix. Find tween of two polygons with same # of points
    - x. In 3d intersect plane or line with