Midterm Review

Computer Graphics
CMU 15-462/15-662, Spring 2016

Notes

- **The midterm is Wednesday, in class. You will have from 10:30am to 11:50am to complete the exam.**
- **No notes, calculators, etc. You should not need them. If you find that you do need a formula, etc., ask me and I'll write it on the board.**
- **These slides summarize the "What you should know" slide at the end of every lecture.**
- **You can find more practice questions, along with solutions, at the Fall version of the course web site. Note that our lectures did not exactly track those of the Fall semester, so these quizzes may include topics we have not covered and we may have covered topics not represented there. However, it is a great practice resource! http://15462.courses.cs.cmu.edu/fall2015/exercises**

Things you should know

Lecture 1: Course Intro.

- For any given setup where we place a camera in the environment, pointing down any of the main coordinate axes (x, y, or z), computing a projection of points in the world onto an image plane.
- Write an algorithm for drawing lines that handles all edge cases (i.e., including edges that are exactly horizontal or vertical).

Lecture 2: Drawing a Triangle / Sampling

1. How should we choose the correct color for a pixel? There is not an exact right answer. However, you should be able to discuss some of the issues involved.
2. What is aliasing, and what artifacts does it produce in our images and our animations?
3. One form of aliasing is where high frequencies masquerade as low frequencies. Give an example of this phenomenon.
4. Suppose we have a single red triangle displayed against a blue background. Does this scene contain high frequencies?
5. What does the Nyqvist-Shannon theorem tell us about how image frequencies relate to required sampling rate?
6. The practical solution on your graphics card for reducing aliasing (i.e., for antialiasing) is to take multiple samples per pixel and average to get pixel color. Try to use what we learned about sampling theory to explain as precisely as you can why taking multiple samples per pixel can reduce aliasing artifacts.
7. Be able to write an implicit representation of an edge given two points.
8. Be able to use the implicit edge representation to determine if a point is inside a triangle.

Lecture 3: Math for Graphics and Transforms

1. Express points and vectors using homogeneous coordinates.
2. Perform a dot product and demonstrate how to use the dot product for projection (e.g., projection of a point onto a coordinate axis).
3. Perform a cross product and demonstrate how to use it to compute a surface normal.
4. Perform matrix multiplication.
5. Derive a parametric expression for a line between two points.
6. Prove that your parametric expression is correct. Discuss whether it is unique.
7. Derive an implicit expression for an edge between two points.
8. Prove that your implicit expression is correct. Discuss whether it is unique.
9. Use an implicit edge expression to determine whether a (2D) point is inside a (projected) triangle.
10. Create an algorithm to rasterize a triangle from a set of "inside-triangle" tests.
11. Make that algorithm efficient so that not every pixel needs to be tested.

Lecture 4: More Transforms

- Create 2D and 3D transformation matrices to perform specific scale, shear, rotation, reflection, and translation operations
- Compose transformations to achieve compound effects
- Rotate an object about a fixed point
- Rotate an object about a given axis
- Create an orthonormal basis given a single vector
- Understand the equivalence of [x y 1] and [wx wy w] vectors
- Explain/illustrate how translations in 2D (x, y) are a shear operation in the homogeneous coordinate space (x, y, w)

Lecture 5: Projections

- Form an orthonormal basis
- Create a rotation matrix to rotate any coordinate frame to xyz
- Create the rotation matrix to rotate the xyz coordinate frame to any other frame
- Rotate about axis w by amount theta
- Know basic facts about rotation matrices / how to recognize a rotation matrix
    - Rows (also columns) are unit vectors
    - Rows (also columns) are orthogonal to one another
    - If our rows (or columns) are u, v, and w, then uXv=w
    - The inverse of a rotation matrix is its transpose
- Create a projection matrix that projects all points onto an image plane at z=1
- Propose a projection matrix that maintains some depth information
- Understand the motivation behind the projection matrix that projects the view frustum to a unit cube
- Be able to draw / discuss the details of the view frustum

Lecture 6: Barycentric Coordinates

- Interpolate colors using barycentric coordinates
- Compute barycentric coordinates of a point using implicit edge functions
- Compute barycentric coordinates of a point using triangle areas
- Estimate the location of a point inside a triangle given its barycentric coordinates
- Estimate the location of a point outside a triangle given its barycentric coordinates
- Estimate barycentric coordinates of a point from a drawing.
- Show that interpolation in 3D space followed by projection can give a different result from projection followed by interpolation in screen space. In other words, explain why interpolation using barycentric coordinates in screen space may give a result that is incorrect.
- How, then, can we obtain a correct result using interpolation in screen space?

Lecture 7: Texture

- Textures are used for many things, beyond pasting images onto object surfaces.
    - Normal maps (create appearance of bumpy object on smooth surface by giving false normal to the lighting equations)
    - Displacement maps (encode offsets in the geometry of a surface, which is difficult to handle in a standard graphics pipeline)
    - Environment maps (store light information in all directions in a scene)
    - Ambient occlusion map (store exposure of geometry to ambient light for better representation of surface appearance with simple lighting models)
    - Can you think of / discover others?
- Know how to interpolate texture coordinates
- Know how to index into a texture and compute a correct color using bilinear interpolation
- Be able to create a mipmap and store it in memory
- Be able to compute color from multiple levels of mipmaps using trilinear interpolation
- What is the logic behind selecting an appropriate level in a mipmap?
- What can happen if we select a level that is too high resolution? too low resolution?

Lecture 8: Graphics Pipeline

- What is the depth buffer (Z-buffer) and how is it used for hidden surface removal?
- Where does the depth for each sample / fragment come from? Where is it computed in the graphics pipeline?
- Is the depth represented in the depth buffer the actual distance from the camera? If not, what is it?
- What is the meaning of the alpha parameter in the [R G B a] color representation?
- Be able to use alpha to do compositing with the "Over" operator.
- Is "Over" commutative? If not, create a counterexample.

$$C = \alpha_B B + (1 - \alpha_B)\alpha_A A$$

- What is premultiplied alpha, and how does it work?
- Be able to use premultiplied alpha for "Over" composition.
- Why is premultiplied alpha better?
- How do we properly render a scene with mixed opaque and semi-transparent triangles? What is the rendering order we should use? When is the depth buffer updated?
- Draw a rough sketch of the graphics pipeline. Think about transforming triangles into camera space, doing perspective projection, clipping, transforming to screen coordinates, computing colors for samples, computing colors for pixels, the depth test, updating color and depth buffers.

Lecture 9: Introduction to Geometry

- **List some types of implicit surface representations**
- **What types of operations are easy with implicit surface representations?**
- **List some types of explicit surface representations**
- **What types of operations are easy with explicit surface representations?**
- **What is CSG (constructive solid geometry)? Give some examples of CSG operations.**
- **What type of representation is best for CSG operations?**
- **Describe how to do union, intersection, and subtraction of geometry using simple operators on a surface representation.**
- **What is a level set representation? When is it useful?**
- **What types of splines are common in computer graphics?**
- **Why are they popular? What properties make them most useful?**
- **Be able to use an expression such as $u^3p_0 + 3u^2(1\text{-}u)p_1 + 3u(1\text{-}u)^2p_2 + (1\text{-}u)^3p_3$ or its matrix equivalent to understand properties of a cubic spline or sequence of cubic splines. Think about continuity at the join point between two splines, whether it interpolates its control points, whether it has the convex hull property.**
- **What was one motivation behind developing the rational b-spline? (representing conics)**
- **What is one advantage of using a *non-uniform* rational b-spline? (control point "strength")**

Lecture 10: Curves, Surfaces and Meshes

- How to use split and average operations to do subdivision
- What is a manifold surface?
- Distinguish manifold from non-manifold surfaces
- Can a manifold surface have a boundary? Give an example.
- Explain the idea of surface curvature with a diagram.
- Give an example of a surface where one of the principal curvatures is zero
- What do you need to store in a halfedge data structure?
- How can you find all vertices in a face with this data structure?
- How can you find all faces that contain a vertex with this data structure?
- Be able to perform edge flips, edge splits, and edge collapse with this data structure.
- BONUS: Think of an algorithm to traverse every face in a manifold using this data structure.

Lecture 11: Geometry Processing

- List practical applications that you can relate to for good geometry processing algorithms.
- List some rules of thumb for good quality meshes
- How can we test if a mesh is manifold?
- Give examples where edge flip and edge collapse may make a manifold mesh into one that is no longer manifold
- Give pseudocode for one iteration of Loop subdivision
- Explain the important properties of various subdivision algorithms (interpolation, continuity, behavior at the boundaries)
- Be prepared to calculate vertex updates in a simple example of Loop or other subdivision. (The vertex weighting masks will be given to you.)

Lecture 12: More Geometry Processing

- Express distance from a plane, given a point on the plane and a normal vector
- Show how the Q matrix (the quadric error matrix) represents squared distance from a plane
- If we have a Q matrix and a point, how do we calculate cost?
- How does this cost represent distance to the original surface?
- Given Q matrices encoding triangles in a mesh, how do we get the Q matrix for each vertex?
- If we collapse an edge, what is the Q matrix for the new vertex that is added in the edge collapse?
- Describe some techniques for improving the quality of a mesh to make it more uniform and regular.