CS 410, Fall 2019, First Third of Semester Review

September 24, 2019

The following is not an exhaustive list of what has been covered, and to be very explicit, the First Midterm may cover material discussed in class not been mentioned below. That said, hopefully the following will be a helpful as you review the broad topics and techniques we have covered so far in CS410.

1. The mathematics of 3-D modeling, both of objects and cameras, rests squarely upon a solid working understanding of the following 3 terms: scalar, vector and point. Your understanding of computer graphics at this time in the course means you're able to succinctly define each of these 3 mathematical concepts.

2. You have now seen how basis vectors play a truly fundamental role in how we express the geometry of object models and cameras. Here's a thought question you should be ready to tackle. In order to span a space, is it absolutely essential that basis vectors be orthogonal? Another way to think about this question is as follows, if it is not a requirement that basis vectors be orthogonal, why does it appear that we place such a high premium on selecting orthogonal basis vectors in practice.

3. You are now aware that there are 2 rather distinct ways of introducing the concept of rotation in the 2-D plane. Most references (e.g. Wikipedia) simply state the rotation matrix in terms of the cosine and sine of an angle theta. There is nothing incorrect about this approach, but it does not really help one understand what is actually taking place when a 2-D point is rotated. Fortunately, you can now take your knowledge of the dot product and derive a 2-D rotation matrix without ever having to touch a cosine or a sine.

4. There are many ways to motivate homogeneous coordinates, fortunately you are now in a position to quickly provide a pragmatic motivation in terms of chaining together, or perhaps one should say composing, sequences of transformations. Further, when making this argument to another it is best to illustrate using concrete examples of commonly named types of 2-D transformations.

5. Homogenous coordinates introduce an ambiguity. There are an infinite number of ways to designate a 2D point (x,y) using homogeneous coordinates and only one of these is canonical. Given any sample from this infinite range of possibilities you may easily now re-express a 2D point in canonical form.

6. It goes without saying that at this point in the course you can write down examples of 2-D rotation, translation, scaling, non-uniform scaling, and sheer matrices - all in homogeneous coordinates. Just as important, if you are presented with a set of 2-D points drawn before and after the application of one or several of these transformations, you can now work out by looking at the relative positions of the points pre- and post-transformation which operation or combinations of operations were carried out.

7. Perhaps one the odder aspects of learning computer graphics is the need first to learn about Euler angles and subsequently to learn why in practice they are virtually never the best way of capturing the rotation of an object or a camera. Fortunately, you can both explain what Euler angles are and why they are not heavily used.

8. If you are asked over lunch in a job interview to compute the cross product between the vectors (1, 2) and (2, 3), be certain that you would know why your reaction should be one of puzzled amusement, aware of the fact that you are being baited into possibly revealing a basic misunderstanding of geometry.

9. There are 2 ways you should have the cross product committed to memory. The 1st is to be able to explain in English the geometric interpretation of the resultant vector. The 2nd is the algebraic approach in which you can actually compute the cross product of 2 vectors.

10. If one were to say there's something special about the cross product between 2 unit-length and orthgonal vectors, you fortunately could quickly explain what is special about this case.

11. In 3 dimensions as opposed to 2 dimensions, it is a bit harder to make up matrices that represent valid 3-D rotations. That said, it is not all that hard, and you now know how. You should be able to explain by example the rules governing the construction of a valid rotation matrix.

12. Just as computer graphics folks don't typically like Euler angles, many of us have a fondness for the axis angle approach to specifying 3-D rotations. Fortunately, you can now illustrate this by showing how to rotate 45° about the vector (2, 2, 1).

13. The geometric concept of a frustum is important to camera modeling. You are now able to explain this geometric construct in your own words and with your own simple drawings. Doing so includes the critical concepts of near and far clipping planes. You are also aware that there is some redundancy in how cameras are specified. For example, an image plane doubled in size and moved twice as far from the focal point results in rendering the same image.

14. In ray tracing the first major formulation shoots one ray per pixel. As introduced in CS410, this means it is essential to be able to enumerate in 3D world coordinates the 3D position of individual pixels. Given a camera specification, you can derive the necessary formulas and associated code to accomplish this task.

15. Also using a camera specification, you understand and can explain the process of generating rays, one per pixel, given a camera specification.

16. You can inspect a camera specification and answer basic questions. For example, comparing two specifications, which will make an object appear larger.

17. In parameterizing a camera, the up vector plays a critical role in selecting the orientation of the camera. Be quick to understand the 'gravity' of the situation – and also this terrible joke.

18. There is a kind of word problem you can now easily solve: a space ship leaves earth in a direction U, how far has it traveled when it passes closest to a planet at position P. Assume the earth is the origin.

19. You now have, if you did not already, a basic familiarity with the characterization of visible light in terms of a spectrum where the wavelengths vary between 380 nm to 780 nm.

20. There are several interrelated ideas that you now can explain to a newcomer all related to the single key term "trichromaticity".

21. Consider the following statement: "The color of light is something very different to a physicist or a hyperspectral sensor than it is to a human eye." You should be prepared to give a concrete example of how this statement is true.

22. One of the most fundamental practical concepts regarding color goes by the name "RGB Cube". Fortunately, you now completely understand what this term is describing.

23. As useful as the RGB cube is, it is not always the best way to think about color. An alternative color space is HSV. Having learned what HSV represents and how it relates to the RGB cube you, should be prepared to answer questions that relate to mappings from one space to the other.

24. Colors are often specified using hexadecimal numbers. For example, #800080 is purple. You should be comfortable answering questions about colors where the colors are specified in this notation.

25. You are comfortable with questions of the style: "Which of these 2 colors is more saturated?" Or alternatively, "Which of these 2 colors appears brighter?"

26. The use of an alpha channel to blend the color of one object with that behind it has now been carefully reviewed and you should be comfortable answering questions about alpha blending.

27. Three kinds of reflectance lie at the heart of computer graphics: ambient, diffuse (Lambertian), and specular. At this point in the semester you have a clear understanding of two: ambient and diffuse (Lambertian) illumination.

28. For diffuse, i.e. Lambertian, illumination you understand how to compute the illumination on a point on a surface given a light source as well as material properties for the surface normal.

29. Calculating where a ray and sphere intersect, and determining if they intersect, is an excellent time in computer graphics to introduce the general technique of mixing parametric and implicit forms in order to solve geometric problems. Consequently, you now are able to use the brute force approach to ray sphere intersection to illustrate in concrete terms this general approach.

30. There is a clever way of determining the intersection of a sphere and a ray. Since you have already implemented this as part of your programming assignments, you're aware that solving a general problem without the aid of a calculator would be annoying. That said, the construction of the algorithm is elegant and important, and you certainly could work out a concrete numerical example if the constituent components of the example were well enough chosen to make the ensuing arithmetic straightforward.