CS559 Computer Graphics – Fall 2015

Practice Final Exam

Time: 2 hrs

1. [$XX \times YY\% = ZZ\%$] MULTIPLE CHOICE SECTION. Circle or underline the correct answer (or answers). You do not need to provide a justification for your answer(s).

   (1) Does a curve that has been arc-length parameterized actually *look* different than one that has not?
   (Circle or underline the ONE most correct answer)

   (a) Yes. An arc-length parameterized curve is clearly smoother than one that has an arbitrary parameterization.

   (b) Maybe. It could look different, if the curve is not C(1) continuous.

   (c) No. A curve can be converted from an arbitrary parameterization to arc-length, and vice versa, with no change to the apparent shape of the curve itself. It would take a dynamic behavior (e.g. observing the motion of an object along the curve) to visually appreciate the difference between the 2 parameterizations.

   (2) Which of the following statements are true about cubic B-splines?
   (Circle or underline ALL correct answers)

   (a) Cubic B-splines always interpolate their control points.

   (b) With cubic B-splines we can obtain C(2) continuity while retaining local control.

   (c) Cubic B-splines will, by construction, never exhibit tangent discontinuities (e.g. sharp corners).

   (3) Which of the following statements about mipmaps are correct?
   (Circle or underline ALL correct answers)

   (a) Mipmaps help eliminate aliasing, while only requiring a minimal memory overhead to store a hierarchy of texture resolutions.

   (b) A disadvantage of mipmaps is that discontinuities are very easily visible when a texture transitions between two different resolutions of the mipmap.

   (c) We can get a very comparable effect to the use of mipmaps by rendering with a standard texture and selectively blurring the resulting image at locations where the texture resolution is significantly finer than the image pixel resolution.

2. [$XX \times YY\% = ZZ\%$] SHORT ANSWER SECTION. Answer each of the following questions in no more than 1-3 sentences.

   (a) Write the stencils for the 2D image filters that would perform the following functions: (i) Blur the image (you may assume a 3×3 window), (ii) Detect sharp vertical edges, and (iii) shift the image up by 2 pixels, and to the right by 1 pixel.

   (b) Although most display devices use primary colors that are *not* pure-frequency colors (i.e. the primary colors span an entire distribution of frequencies), this does not significantly compromise the ability of such devices to reproduce a reasonably extensive color gamut. Why is this so? How can two monitors produce the "same" color even if they use different primaries?

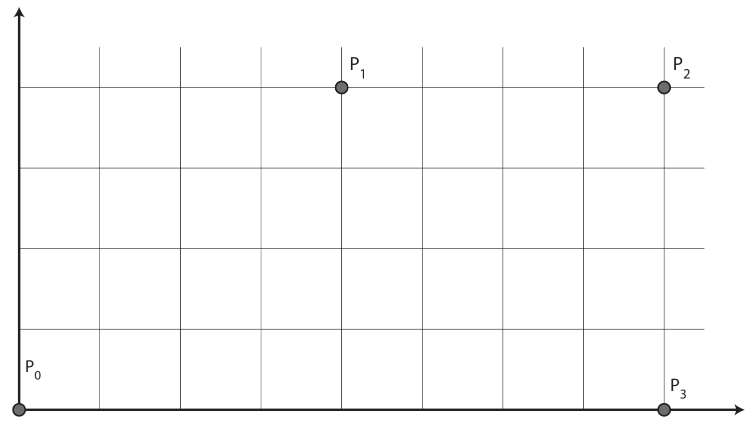
   (c) Name two reasons why, in ray tracing, we may find it preferable to send rays *away* from the camera and into the scene, rather than sending out rays from light sources and *towards* the camera.

(d) Explain why a curve that is $G(1)$ continuous *and* arc-length parameterized will necessarily be $C(1)$ continuous, too.

(e) In local illumination models (e.g. the Phong model we discussed in class) shading of a given point on a visible surface object is computed independently of how any other surface has been shaded. Describe two examples of scenes or phenomena that a local illumination model would not be able to realistically reproduce.

(f) If $\mathcal{C}(t) = \begin{pmatrix} x(t) \\ y(t) \\ z(t) \end{pmatrix}$ is a curve in 3D, show that the vector $\begin{pmatrix} y'(t)+z'(t) \\ -x'(t) \\ -y'(t) \end{pmatrix}$ is always *perpendicular* to it *[Hint: How does it relate to the tangent?]*

3. [XX%]Consider the four points $\mathbf{P}_0(0,0)$, $\mathbf{P}_1(4,4)$, $\mathbf{P}_2(8,4)$, $\mathbf{P}_3(8,0)$ as shown in the illustration below:



Let $\mathcal{C}(t)$ be the cubic Beziér curve with control points $\mathbf{P}_0, \mathbf{P}_1, \mathbf{P}_2, \mathbf{P}_3$. Use De Casteljau's algorithm to compute the intermediate curve locations $\mathcal{C}(0.25)$, $\mathcal{C}(0.5)$, $\mathcal{C}(0.75)$ (on paper), and use them as guides to draw an approximation of the curve on top of the figure above.

It is also **acceptable** to compute the intermediate curve values graphically, instead of computing exact coordinates, if you prefer doing so. If you choose this option, you might want to show your work separately for each intermediate point (in a duplicate of the figure above) to avoid clutter, and then transfer your final results to the illustration above.

4. [XX%] Consider a *quadratic* parametric curve $\mathcal{C}(u) = \mathbf{a}_0 + \mathbf{a}_1 u + \mathbf{a}_2 u^2$ in 3 dimensions, which satisfies the following constraints:

   - $\mathcal{C}(0) = \mathbf{p}_0$
   - $\mathcal{C}''(0) = 4(\mathbf{p}_0 - 2\mathbf{p}_1 + \mathbf{p}_2)$
   - $\mathcal{C}(1) = \mathbf{p}_2$

   where $\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2$ are control points given as input.

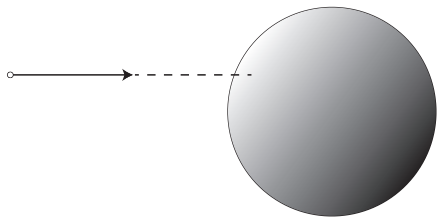
   Compute the basis matrix $\mathbf{B}$ that would allow us to express this curve in the form

$$\mathbf{C}(u) = \underbrace{\begin{pmatrix} 1 & u & u^2 \end{pmatrix}}_{\mathbf{u}} \mathbf{B} \underbrace{\begin{pmatrix} \mathbf{p}_0 \\ \mathbf{p}_1 \\ \mathbf{p}_2 \end{pmatrix}}_{\mathbf{P}} = \mathbf{uBP}$$

5. [XX%] Consider a ray, originating from point $\mathbf{p}_0 = (2, -3, -1)$ and propagating in the direction of the unit vector $\mathbf{d} = (0, 1, 0)$.
[Hint: The parametric equation of the ray will be $\mathcal{C}(t) = \mathbf{p}_0 + t \cdot \mathbf{d}$, $t > 0$]
A sphere with equation

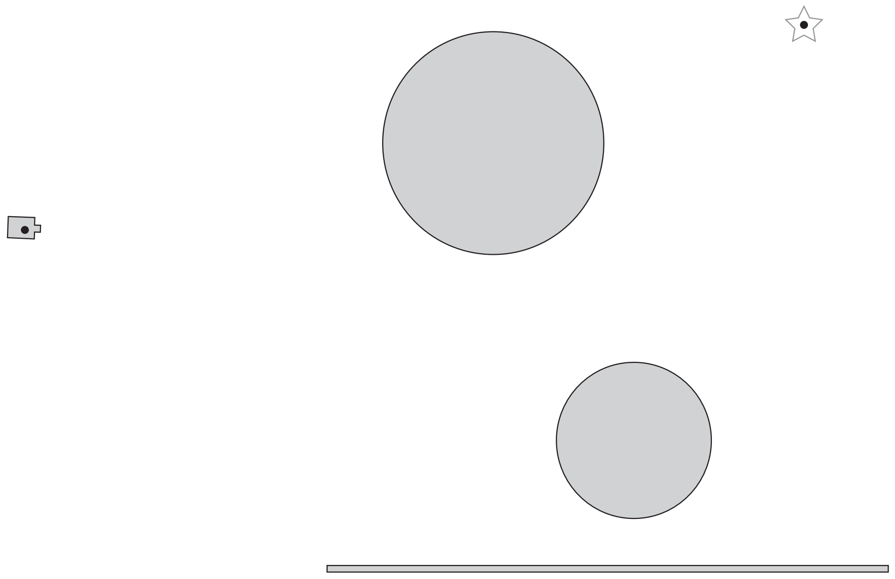
$$x^2 + y^2 + z^2 = 9$$

(centered at the origin, with radius 3) potentially lies in the path of the ray, as illustrated below.

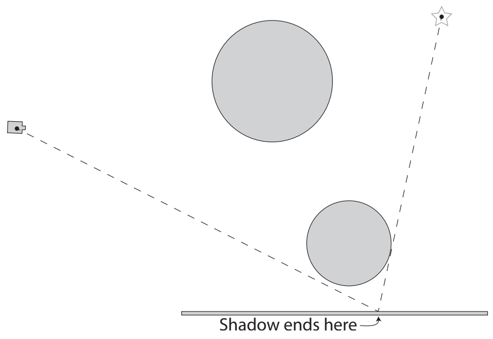


Compute the point where the ray intersects the sphere, and also compute the surface normal (relative to the sphere surface) at the point of intersection. If you believe that the ray and the sphere do not intersect, make sure to justify your claim.

6. [XX%] In recursive ray tracing, we cast rays from the camera and out into the scene. When a ray intersects an object, it can generate a reflection ray (in the mirror-reflection direction), as well as shadow rays (for each light source, and pointed towards it). For this scene, we assume we cannot have the third type (transmission or refraction rays) since all objects are non-transparent. Assume that our algorithm allows each ray to bounce *no more than two times* off objects, before reaching a light (rays that cannot reach a light after a maximum of 2 bounces are eliminated).



In the 2D scene above, we have a camera (left), a point light (drawn as a star; top right), two sperical objects and a planar surface on the bottom. (i) Show which parts of the object surfaces will be viewed as illuminated, and which ones will be shadowed (draw directly on the figure above). (ii) Draw *one* sequence of rays with **two** bounces that reaches the edge of one of the shadows. (For example, the illustration below shows a one-bounce ray path that reaches the edge of a shadow). Indicate any equal angles.


Shadow ends here