GENERAL (Answer 4 of 6)

Counting holes

I have a triangle mesh $M$ with $t$ triangles and $v$ vertices. It is edge-connected and forms a manifold with boundary (zero or more manifold border loops). $b$ corners have no opposite. How may border loops does $M$ have? Provide a formal proof that justifies your answer (give references or justifications of each step of your derivation).

Counting components

I have a manifold triangle mesh with $T$ triangles that is made of a large number of pairwise disjoint components (each having no more than $k$ triangles). I want to paint each component with a different color. The only representation I have is the triangle/vertex incidence $V$ table. I do not have and do not want to build the Opposite table, nor any other adjacency graph. I cannot use geometry for this process. Provide a simple algorithm to compute the color of each triangle. (You may use some book-keeping bits for this process). Discuss the computational complexity of this algorithm.

Mipmaps

This question is about Lance Williams' mipmap technique for anti-aliasing textures.

a) How does a mipmap handle texture magnification, when the texture is so close that one texel would cover many pixels in screen-space?

b) How does a mipmap handle minification, when many texels are compressed into a single pixel in screen-space? You may want to use the idea of a "pixel footprint", the area of a texture that falls within a given pixel.

c) Mipmaps are poor at handling texture maps that are viewed nearly edge-on, that is, when the texture footprint is long and thin. Explain why mipmaps are poor in this case, and propose a potential modification or variation of mipmaps that can help fix this problem.

Motion

You are given three points $A$, $B$, and $C$ and asked to produce a smooth motion of a point $P(t)$ such that $P(0)=A$, $P(1)=B$, and $P(2)=C$. Explain how you do that. Discuss the mathematical nature of the motion (is it a known curve?), its smoothness properties, how you would represent it in a computer, how you would derive $P(t)$ from that representation, and give the precise formulae or geometric construction for computing that representation.

Warp

The artist is given a polygonal curve $P$ that densely samples a smooth curve $C$. To deform it the artist draws a short polygonal curve $Q$ close to a portion of $P$. The curve is to act as an attractor. Explain how you would compute the deformed $Q$ so that it remains smooth but satisfies the artist. Give construction details. Comment on what could go wrong.
**Parametric curves**

In this problem, we will explore the construction of parametric.

a) Given the following Bezier control points, construct all of the de Casteljau lines and points needed to evaluate the curve at \( u = 1.3 \). Mark this point on your diagram and then sketch the path the Bezier curve will take. The curve does not need to be exact, but it should conform to some of the geometric properties of Bezier curves (convex hull condition, tangency at endpoints).

![Bezier control points]

b) Given the following de Boor points, construct all of the lines and points needed to generate the Bezier control points for the B-spline. Assume that the first and last points are each repeated three times, so that the spline is endpoint interpolating. You must mark each Bezier point (*including any that coincide with a control point*) with an X, but you do not need to label it (i.e., no need to give each Bezier point a name). Sketch the resulting spline curve, respecting the properties of Bezier curves noted above.

![de Boor points]

c) Given the following Catmull-Rom control points, construct all of the lines and points needed to generate the Bezier control points for the Catmull-Rom curve. Use a tension value of \( \tau = 1 \). Assume that the first and last points are each repeated two times. You must mark each Bezier point (*including any that coincide with a control point*) with an X, but you do not need to label it (i.e., no need to give each Bezier point a name). Sketch the resulting spline curve, respecting the properties of Bezier curves noted above.

![Catmull-Rom control points]
MODELING (Answer 2 of 4)

Clean-up

To draw the boundary of a finite, simply connected, manifold 2D region $R$, the lead artist draws a set $S$ of $n$ strokes. For simplicity, we assume that each stroke is a line segment represented by its two vertices and that no 3 vertices are collinear. However, a stroke may intersect other strokes. Furthermore, the union of all strokes may not necessarily form a connected set that completely encloses $R$ (there may be small gaps between connected lumps of stroke patches as you walk from one patch to the next around the boundary). The job of the clean-up artist is to draw a manifold, closed loop curve around the guessed region $R$ so that the curve is fitting the strokes. Suggest an algorithm that would replace the clean-up artist and produce a polyloop around the guessed $R$: (1) Use mathematical, topological, morphological… terms to describe at a high level the essence of the mapping from $S$ to $R$ that your algorithm performs, (2) List and describe the steps of the algorithm at a level that would give an expert developer in geometry processing sufficient guidelines to implement them, (3) Discuss situations when your algorithm may produce results that are surprising to the clean-up artist.

Vertex soup

Let $R$ be a regularized and manifold planar region with axis-aligned edges. (1) Explain at a high level how to derive the edges of $R$ from the set of its vertices (listed in random order). (2) Justify the validity of your approach. (3) Can your approach be extended to 3D (where $R$ is a regularized and manifold polyhedron with axis-aligned faces)? (4) Justify your answer with a formal proof or counter-example.

Skin

Consider the surface $A$ that bounds an artery and a stack of 4 parallel cross-sections roughly orthogonal to the centerline of the artery. In each cross-section $k$, you are given an approximation of the intersection of $S$ (with the cross-section) represented as the control points $P_{k,j}$ (for $j$ in $\{1…n_k\}$) of a cubic B-spline curve $C_k$. We want a $C^2$ continuous surface $S$ that interpolates the four curves $C_k$ and is a good approximation of the corresponding section of $A$. Explain how you would represent $S$ and how you would derive its representation from the control points $P_{k,j}$. To make your explanation clear, decompose your approach into steps and provide high-level, but sufficient implementation details of each step. Argue why the resulting surface is $C^2$.

Hiding place

The ruins of a Greek temple consist of a large number $c$ of positioned cylindrical columns, each one of a different color, all of the same radius $r$, randomly positioned on a disk platform of radius $R$. The
guard can walk around the temple, but cannot enter the platform. His job is to inspect the temple every morning and count the columns. Given a 3D model of the temple, how would you test whether the guard can do his job (i.e., whether he can actually see at least a bit of each column from some point outside of the platform). We want an exact algorithm (not one based on sampling). Explain the essence of your approach, provide some algorithmic details, and analyze its computational complexity.
VR/AR (Answer 2 of 4)

Interaction Techniques

Consider the typical VR environment, where people sit in one place with a head-worn display on and use devices to control their motion through the environment (travel) and their interaction with the environment (manipulation).

a) There are two classes of 3D interaction techniques, roughly classified as "ray-casting" and "image plane" techniques. Define these techniques, give specific examples of both classes of techniques (preferably more than one each), and discuss the kinds of interactions they are well suited for (and not well suited for).

b) How does your discussion change if you consider CAVE environments, instead of HMD-based environments?

c) How does your discussion change if you consider video-mixed HMD-based AR environments, instead of HMD-based VR?

d) How does your discussion change if you consider optical-see-through HMD-based AR environments, instead of HMD-based VR?

Collaboration

Consider a collaborative virtual environment, where one person is in a VR based interface and one person is in AR. Such an environment might be used to allow a remote person (VR) to interact with someone who is at some location (such as a repairman, or soldier, or perhaps even someone taking part in an entertainment application). A major problem with such remote collaboration is latency. Different kinds of environments, such as MMOs (e.g., SecondLife, Everquest) and online games (e.g., Quake, Halo, Left4Dead), have come up with ways of dealing with these problems. The solutions differ depending on the scale of the environment, and the particular tradeoffs that are important for that environment.

a) There are three major kinds of approaches to dealing with scale and latency, which are exemplified by environments like SecondLife, World of Warcraft, and games like Unreal Tournament. Describe the differences in these approaches, why they were taken, and what the pros and cons of each are.

b) If one of the "players" in each of these environments is in AR, does it change things? Describe (for each) if AR changes the requirements, and outline how you would account for this. Which of these is best suited for AR/VR collaboration?

c) Are there any other issues that arise when using typical VR/games algorithms to create a system that allows players to interact across these two different environments? Please summarize what these differences are, if any. If there are no significant differences, say why you think there are no differences.
Tracking

Assume you have a video mixed HMD, similar to the one used in the "AR Pit" experiment. We could track the user in two ways: (a) an external tracking device, such as the IS-900, or (b) computer vision, using the camera on the HMD.

(a) Describe the possible sources of error for each approach. The error we care about is the registration error at each rendered frame.

(b) Each method has pros and cons besides just the registration error. Discuss them.

Displays

Consider the different ways of displaying information in Augmented Reality (optical see-through displays, head-worn versus handheld displays, projection versus worn).

(a) Compare head-worn displays to handheld displays for mobile AR. What are the pros and cons for each, from a technical, application and user perspective?

(b) Many people imagine optical see-through handheld displays. Describe the technical challenges in make such a display work, including both display, application and usability issues (you don't have to solve them, just define them and discuss them!).
RENDERING (Answer 2 of 4)

Underwater Film

You have been hired to create the rendering engine for a feature film that takes place entirely underwater. The director has asked you to create a renderer that will produce effects including 1) distortion of objects outside the water when viewed through the air/water interface, 2) caustic light patterns on the shallow ocean bottom from focused light at the surface, and 3) light shafts created by reflected particles in the water.

a) Your first design for your system is ray-tracing based. Describe how each of the three effects can be created.

b) The director got your rendering time estimates for your ray tracer from part (a) and asked you to re-do the entire system using less computationally costly approaches. How will you produce the effects that the directors need, using faster (but perhaps approximate) methods?

Architectural Lighting

You have been hired by an architectural firm in order to write their rendering engine. (They are friends of the director of the underwater film.) They work entirely with diffuse surface interiors, but they are very particular about capturing diffuse/diffuse light reflection in their virtual building interiors.

a) Describe a radiosity approach for creating their rendering system.

b) Describe a Monte Carlo ray tracing method for producing the kinds of scenes that they want.

c) With unlimited computational resources, will both of the approaches converge to the same final images? Explain why or why not.

d) Both of the above methods may have visual artifacts (e.g. visible seams between patches, noisy shading). For the renders each of in (a) and (b), describe these artifacts and tell how you would use additional computational resources to improve the images. Come up with a quantitative measure for the quality of the results, and describe how this quality changes with more compute resources. Try to provide equations for this cost/quality relationship.

Stripes

You are given the task of writing a procedural shader (a solid texture) that makes 3D objects look as if they are carved out of a black-and-white striped solid material. That is, the 3D volume is as if large slabs of alternating black and white solid blocks have been placed one on top of the other.

a) Describe (using psuedo-code or equations) how to create such a procedural shader. Do not use a texture map or a voxel map.

b) When the stripes of your shader are on a distant object, your shader will alias considerably. Propose a super-sampling method of anti-aliasing your shader. How will you create your samples, and how will you blend them together to get the best quality?
c) Supersampling is expensive. Describe a method of modifying your shader (code or equations) that reduces the aliasing problem without super-sampling. You may use the size of the pixel footprint as part of the input to your shader.

Cylindrical Light Fields

Light fields, as described by Levoy and Hanrahan, are given by a 4D set of samples in a “light slab” that are indexed according to rays passing through two parallel planes, the (u,v) and the (s,t) plane. Although one light slab is fine for viewing one side of an object, it has the drawback that you cannot walk all the way around an object using just a single light slab. Consider modifying the light field approach so that you index rays according to two nested cylinders, instead of two planes. Positions on each of the cylinders are described by cylindrical coordinates, of course, with the two coordinates being angle around the circumference and height in the direction parallel to the axis. Since you use two cylinders, one inside another with a shared axis, you still have a 4D set of samples.

a) Describe how you would use rendering software to create a synthetic cylindrical light field for a model of a dinosaur. Give details of how you would capture each sample for this 4D representation.

b) How would you use a cylindrical light slab to create a synthetic image of the dinosaur from an arbitrary view position? Be sure to consider the problem of aliasing and address this issue in your answer.

c) Are there any drawbacks that you can see of cylindrical light fields, such as computational costs, quality of rendering, restriction on view positions, and so on?

d) Is a cylindrical light field the same as the Concentric Mosaics representation of Heung-Yeung Shum from SIGGRAPH 1999? If not, how do the two methods differ?