You must answer ALL **6** questions.

Each question has multiple parts. Roughly, each question part is weighted the same (so question 3, with 4 parts will be weighted more than question 6, which only has one).

For full credit, an answer must be both correct and well-presented (clear and concise).

If you feel a question is ambiguous, state any assumptions that you need to make. Hint: more often than not, this is a sign that you either do not understand the question, or are missing some important insight or piece of knowledge.

Several of the questions are "essay" questions. For these essay questions, there are many correct answers. It is more important that you provide a good argument for the answers you give than that you give the ``most correct possible" answer. Sometimes we are specifically looking for your ability to make a clear and concise argument based on things you are aware of, rather than to see if you can find the best possible answer or have seen all possible research on the topic.

Question 1: Quicktime VR, Sampling, Graphics Hardware, Textures

In the early 1990s, Apple introduced Quicktime VR (QTVR): a technology that allowed viewers to navigate panoramic imagery. At the time, there was no consumer-level 3D graphics hardware; thus Quicktime VR was one of (if not the) first successful technology for giving average users an "interactive 3D experience."

In the years since, computing hardware has progressed significantly. Almost all computers have the ability to display large numbers of texture-mapped 3D polygons. Most people have high-bandwidth network connections, and the cost of data storage is a tiny fraction of what is was back then.

[1a] In terms of its high-level goals what key ideas from QTVR are still appropriate?

[1b] Suppose 3D scene reconstruction from photographs was possible. What advantages would a QuicktimeVR-like approach have?

[1c] One interesting technical decision in QTVR was to represent the panoramas as cylinders, rather than using cubes. This choice is an historical argument: much of the reasons it was done at the time had more to do with performance issues on the circa-1994 computers it was developed for. Nowadays, performance is not a concern.

Discuss the tradeoffs between Cylindrical and Cubic panoramas if you were to create a Quicktime-VR-like system today.

Question 2: Subdivision, Smoothing, Fairing, Filtering, and Bi-lateral Filtering

There is a considerable literature on creating "smooth" surfaces, and for making polygonal meshes "smoother." However, often the literature prefers to talk about surface "fairness" rather than smoothness.

Bi-lateral filtering was originally an image filtering technique that provides a new meaning for "smoothness," and has wide applicability in many applications. In fact, it has been applied to surface/mesh filtering (although you were not expected to read that paper).

Consider extending traditional surface fairing (e.g. Taubin smoothing) to be a "bi-lateral filter". What would you expect the results to look like (compare to applying a standard Taubin filter)? Why do you think this would be useful?

(Hint: consider using similarity of normal directions as a measure of "closeness" for the bi-lateral weighting).

Question 3: Rendering

Because ray-tracing terminology can be ambiguous, we will be explicit and use the term "from-the-eye" ray casting and "from-the-light" ray casting (rather than forward and backwards).

Nature uses a "from-the-light" ray tracing strategy to form images. So, almost by definition, any natural image can be formed by from-the-light ray tracing. It just may not be efficient (in terms of the number of rays that need to be considered, the number of ray/object interactions that must be considered, etc.).

Computer graphics, in contrast, often uses "from-the-eye" ray-tracing in order to form images using a more reasonable number of rays. However, some effects cannot be created efficiently using a purely from-the-eye raytracer. Some of these cases are handled by doing bi-directional ray-tracing: a class of methods that mix from-the-eye and from-the-light rays.

[3a] What visual effects are difficult (i.e. requiring an impractical number of rays) to achieve using a "from-the-eye" ray tracer?

[3b] A key component of a bi-directional ray-tracer is a data structure that store the results of one direction of ray-tracing so that it can be used by the other direction. Describe how this bi-directional approach can achieve certain image effects using far fewer rays (and ray-object computations) than would be required using just one direction or the other.

[3c] What question (3b) gets at is that bi-directional ray-tracing is effectively a mechanism for trading storage for computation (space for time). By using memory (storing the results from one direction) we save time (we can get by with fewer rays).

What are the effects of scaling the amount of memory used for storing the intermediate results in a bi-directional ray-tracer. Is increasing the amount of memory (e.g. the resolution) of the storage always a good thing? What are the effects of having too much or too little resolution in this intermediate storage?

(Hint: for a given scene, more memory for the intermediate storage means more resolution, and photon maps are one example of a data structure for this intermediate storage)

[3d] With all 3 ray-tracing strategies (from-the-eye, from-the-light, bi-directional), image quality improves as the amount of computation (e.g. the number of rays considered, and the number of ray/object interactions considered). However, in each of the approaches, the

ways in which the image quality would change as the amount of computation increases may be different. For each strategy describe ways that image quality improves as the amount of computation increases. (you may consider this in the negative - as you reduce the number of rays, what kinds of artifacts become visible). Are there differences between the approaches?

Question 4: Motion blur and temporal antialiasing

Motion blur occurs naturally in still or video cameras when fast moving objects are shot with relatively low shutter speeds. While high-frequency camera shake is typically an undesirable artifact in still photography, a consistent and directional motion blur can be an effect that is intentionally sought after. In particular, visible motion blur in photography can be used for artistic effect, conveying a perception of motion and liveliness in an otherwise still picture.

[4a] Video sequences inherently convey motion in the environment. However, in certain instances the absence of motion blur can compromise the resulting video sequence, both in quality and plausibility. Provide an example where the absence of motion blur leads to a video sequence where the **perceived motion is different** from the motion in the real scene being imaged.

[4b] Adding motion blur synthetically (temporal anti-aliasing) is often costly. But in some situations, it can be very important. Discuss the kinds of situations under which temporal anti-aliasing is likely to be important in rendering an animation.

Hint: Are there particular types of motions, scene geometry, or rendering algorithms that are likely to make temporal anti-alising more or less important.

Question 5: Mesh-based deformers for animation

A recent approach to character animation is to enclose the geometry of the character (typically a polygonal mesh) within a simpler polyhedron, called a cage. As the cage is manipulated, the geometry is deformed accordingly.

[5a] What are the advantages and disadvantages of having animators manipulate the character using the cage rather than the surface mesh itself? (Hint: there are good and bad traits in both approaches)

[5b] One way to implement the cage-based deformations is to divide the control polyhedron into tetrahedra, and use barycentric interpolation to position each vertex of the mesh within the tetrahedron that contains it. A second way to implement cage-based deformations is to use some form of generalized Barycentric co-ordinates to define a coordinate system over the entire control polyhedron. Discuss the advantages and disadvantages of each approach. (Hint: there are good and bad traits in both approaches)

[5c] Harmonic coordinates are one popular technique used to create the generalized barycentric coordinates for skinning. What nice properties of Harmonic coordinates make them particularly useful for skinning?

Question 6: Spatial partitioning data structures and geometric queries

Explain how one can use a k-d tree data structure to accelerate the detection of collisions between two triangle meshes that have been rigidly transformed such that they overlap with one another. For the purposes of this question "collision detection" is the enumeration of all intersecting triangle pairs (containing one triangle from the first mesh and one triangle from the second). You may assume that a subroutine that tests 2 triangles for intersection in O(1) time is available to you as a library function (i.e. you do not have to explain how to implement it)..

(Hint: you should consider how a spatial data structure helps in general, as well as why K-d trees in particular are a good choice)