You must answer all questions.

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. Also, several of the questions are "essay" questions. For these, 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" answer. Sometimes, we are particularly 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.

Problem 1 : The modern paper (3 questions)

You were asked to review the paper
"Fast Automatic Skinning Transformations"
by A. Jacobson, I. Baran, L. Kavan, J. Popovic and O. Sorkine,
ACM Transactions on Graphics 31(4) (Proceedings of SIGGRAPH 2012).

1. This paper defines a parametric subspace of shape deformations which is based on the Linear Blend Skinning (LBS) method (with some strategically crafted add-ons such the rotation clusters and additional weight functions). It is known that the main benefit of LBS is simplicity and speed, but at the same time its susceptibility to deformation artifacts is well documented, especially near rotating joints. For the purposes of a pure skinning application (not necessarily in the context of a more elaborate task as presented in the paper), alternatives such as Dual Quaternion Skinning are much more resilient to such deformation artifacts while only incurring a rather modest computational overhead. Could this paper have utilized Dual Quaternion skinning as the basis of its deformation subspace? What advantages or disadvantages would be associated with such a choice?

2. A key idea in the paper is giving an animator the option to only specify a small number of degrees of freedom (namely, the affine transformations associated with just a few of the rig "handles") and for the system to automatically infer the rest via an optimization process. Since the optimization functional is an energy (As-rigid-as-possible) that makes sense for any mesh, not just a mesh that deforms in a parametric subspace, why would we choose to impose a "rig" on the part of the articulated character that we don't care to control via user input? For example, if we are animating a quadruped animal, but we only intend to hand-animate the legs, why would we care to constrain the rest of the body to abide by a (ultimately procedural, not physically-accurate) skeleton/rig whose parameters we do not even care to prescribe? Why not simply minimize the As-rigid-as-possible energy (or any other deformation energy) on those parts of the character we do not explicitly control? Explain the possible merits and/or pitfalls of this line of thought.

3. The method described in this paper is advertised as being able to achieve effects that would be tedious to achieve with previous, simpler techniques. While they compare to linear blend skinning, they do not compare with cage-based deformations. (for example, harmonic coordinates). While such techniques often require expensive pre-computation, they also often can provide real-time posing performance, so they are a valid competitor.

   Describe a demonstration that would provide compelling evidence of a situation where the method of the paper would have clear advantages over a harmonic (or similar) coordinates cage-based skinning technique (think of it as if you are the paper author, and had to add another demo to the video to convince a reviewer. you probably should have effects that would be difficult to achieve with the cage).

Problem 2: Geometry representation and manipulation (2 questions)

1. Implicit surfaces represent the surface of an object as the zero isocontour of a scalar field. Signed distance fields (also referred to as level sets) additionally have the property that the absolute value of the scalar field measures the distance to the object surface. Imagine having two similar, but not identical models, each encoded as an implicit surface. We want to morph between the two, in a smooth fashion, and describe the result as an implicit surface as well. In which instances could we expect simple linear interpolation of the respective scalar fields to be a good solution (or approximation)? Can you describe a scenario where simple linear interpolation of the scalar fields would risk artifacts, or otherwise unacceptable results?

2. You are developing computer modeling software that allows you to design and style digital models of women's dresses, for a clothing company. The design department is providing you with surface models of dresses, corresponding to early prototypes, that they authored using third party software. Their triangulated surface models are given to you simply as a list of vertex triples (i_1, j_1, k_1), (i_2, j_2, k_2), ... (i_n, j_n, k_n) containing the integer vertex indices of each triangle and, additionally, and a list containing the 3D coordinates of all vertices.

   [Note: the specifics of the file format encoding the information above are not significant. What is significant is that you are not explicitly given any additional information such as pairs of adjacent triangles, incident triangles on a vertex, etc. If you need any such information, you should explain how you compute it.]

   A. Prior to inputting this mesh into your pipeline, you want to perform some rudimentary sanity checks that the input mesh is of "good" quality. Among those, you want to ensure that the garment has exactly 4 holes/openings, one at the bottom, one at the neckline, and one at each sleeve. How would you implement that check using only the information provided in the described input format?

   B. Suggest additional sanity checks that you might include, to help ensure the garment model is ready for follow-up modeling, texturing, and possible simulation.

Problem 3 : Color Theory (2 questions)

The so-called "Rainbow Color Map" (RCM) is often used to encode a scalar quantity as a color for data visualization.

Note: while RCM was not part of the qual reading list, the ideas of visual encodings were, as well as perception, particularly of color. While you could answer this question by remembering some of the literature about RCM, you should be able to answer this question based on your understanding of visual encodings and perception

The rainbow color map encodes a continuous value as hue. Sometimes this is done in a truly spectral manner so that the value is mapped to the spectral frequency of the color, although usually the value is mapped around the color wheel. For example, the lowest value is mapped to blue, the highest value is mapped to red, and values in between are interpolated around the color wheel

1. There are many reasons why the RCM is a bad idea for use as a visual encoding of a continuous variable. Describe some of them.

   Note: if you choose "it doesn't work well for color blind people" as one of the reasons (and it is a good reason), you must explain how the RCM would be seen by someone with the common form(s) of color blindness.

2. Consider showing the temperatures in Madison over the course of the year with a color encoding. Give an alternative to the RCM. Discuss the pros and cons of each.

Problem 4 : Filtering and Texturing (2 questions)

MipMaps, combined with tri-linear interpolation, have become the de-facto standard for texture map filtering. By making some simplifying assumptions to the actual filtering problem, they can provide for very efficient performance. Not including pre-computation time to create the MipMaps, each filtered texture access requires a constant number of map accesses.

However, these simplifications cause limitations on the filters that are realized by the method. Phrased differently, MipMapping restricts some of the filtering options that other methods might use to create different performance.

1. Describe some of the simplifications and limitations that are introduced in order to do MipMap + tri-linear interpolation (as opposed to better approximations of the ideal filtering). For each one, explain why this limitation is required to achieve the constant-time performance, and describe a situation where this simplification and/or limitation leads to a visible artifact (it is best to describe this in terms of a picture and what the viewer would see, compared to a more "ideal" method that didn't have the limitation).

   Note: it is better to have 2-3 good limitations, than a larger number of not-so-good ones.

2. MipMaps + trilinear interpolation could be used for implementing the image filtering for other image resampling operations than texture mapping. In fact, they are a pretty effective tool for doing image warping, but probably overkill for more basic image resampling. Why would MipMaps + trilinear interpolation not be an effective tool for:

   A. Image Resizing (scaling an image, potentially a different amount in each direction)

   B. Image Rotation

   Hint: Why do these special cases allow more ideal resampling to be efficient (even without mip-mapping)? Why do the simplifications of mipmapping make it less effective?

Problem 5 : Stylized and Photo-realistic Rendering

Many shading models are intentionally non-realistic (as opposed to models that are just simplified for practical reasons, or poor approximations to reality despite trying). Suppose that someone was able to figure out how to do completely realistic shading in a fast and practical way.

Give examples of current "intentionally non-realistic" (or stylized) shading models that might still be used, and give arguments for why they would be useful in a world where we had the choice of realistic shading models.