Advanced Computer Graphics Summative Assignment - LLLL76

QUESTION ONE

An appearance based metric is centred around using the perceivable difference between two corresponding raster images that are produced by the renderer. The difference can be calculated as the average sum of squared differences between all corresponding pixels, using the euclidean distance between two RGB vectors as representation of the distance between two pixel values. There are more complex methods of difference calculation between two vectors but this works well in this domain. If the difference between the vectors is small then the model is a good representation of appearance in this specific view, a total can be given as the integral over a finite set of viewpoints. The benefits of this method is that similarity of appearance is directly measured and occluded details can be removed without introduction of any error. The problems are that sufficient sampling of the possible viewpoints needs to be done so as to avoid removing perceptually important features, which leads to expensive rendering step sometimes being required, only a reduced number of samples can be taken.

A geometric based metric is based around producing a geometrically faithful representation of the data using techniques derived from function approximation. This does not allow for removal of features that are occluded as this is not known, which leads to an increased number of points being fitted to, however many viewpoints are not calculated as all features are represented.

The appearance based metric should be calculated after a model is displayed, this is towards the end of the pipeline. The geometric based metric should be done after the rendering but before the model is displayed.

QUESTION TWO

The Hausdorff distance can be applied where there are a miss matched number of points in each of the meshes as they are not 'paired up' in such a way that this is necessary. A point from M_1 will calculate the distance to the nearest point in the mesh M_2, this is done in such a way that multiple points from M_1 can measure a distance to a single point inM_2. This is only done within a local area and if no point is found the distance is set at a maximum. This is also done backwards; M_2 points are matched to the nearest M_1 points, the two sets of distances are then used to give a metric for comparability, be that mean of all values or sum of all values.

QUESTION THREE

Providing the same functionality as the simplification stream, meaning that at any point we can reconstruct any intermediate model in the simplification process. The benefits of the progressive over the simplification is that the resulting representation can be smaller than the original model and reconstruction time is proportional to the desired approximation size. The progressive mesh exploits the fact that the contraction operation is reversible and is in essence a reversed simplification stream. For each of the contractions in the simplification stream we define a corresponding inverse, called a vertex split. Along with the base mesh or original model we can store each of these vertex splits, each must encode the vertex being split, positions of the two resulting vertices, and which triangles to introduce into the mesh. Not only being able to encode intermediate steps but also being an efficient compression technique.

Not sure about the rendering efficiency.

QUESTION FOUR

The refinement process data structures stores the data split that occurs for each of the vertices, this allows for each vertex to be split into two, of which the triangles stored as part of the data structure along with the coordinates of the two new vertices fit allowing for greater fitting to the model and higher granularity, this is the refinement process. The decimation process is calculated using the data split, as this is the inverse of the contraction operation, it is a straightforward inverse to calculate the corresponding contraction operation for each of the data splits, this allows for the decimation process to be done.

QUESTION FIVE

Level-of-detail modelling is defined as reducing the storage size of a given model such that the detail of the model is of a pre-specified level. This is done by reducing the number of polygons used to represent distant objects as fine details become less visible at greater distances. This method has the drawback of requiring the whole model to be sent to the client before rendering can be performed, unlike other objecting encoding methods. A progressive mesh is a a method in which a model is decomposed into a base mesh and a sequence of progressive records. The base mesh represents a minimum resolution that the model can be represented by and each of the sequence of progressive records is a node split that will increase resolution of the model. The records can be applied up to any degree within scope to allow for fine grain resolution changes. The data for a given resolution can be requested in an on demand style, allowing for bandwidth usage to be controlled. This is a very suitable for a network that has bandwidth limitations.

QUESTION SIX

A large 3D scene will have obscured faces that LOD modelling does not take into account, this will incur a time cost. Any parts of the scene may be out of the frustum at any given time, LOD does not take this into account and will incur a time cost. If the LOD technique is view-independent then the level of detail across the whole scene will be the same and this fails to take into account the distance factor and will not provide an appropriate level of detail across the whole model, this is the main problem with LOD. The first more basic method for reducing render time is to apply hierarchical culling. This will remove any faces that are out of view and thus reduce the render time. The second method is based on view frustum, surface orientation, and screen-space geometric error, and will develop a real-time algorithm for incrementally refining and coarsening the mesh according to these criteria to allow for variable meshes to be used throughout the model whilst maintaining a smooth visual transition between any two different meshes. This is done through the basic idea of iterating through all active vertices before rendering and either split it. collapse it, or leave it as it is based on the criteria stated before.