Advanced Computer Graphics Summative Assignment

hzwr87

1 Compare the main difference between applying appearance-based metric and geometric-based metric to measure the quality difference between two polygon meshes. Analyse in which part of the graphics rendering pipeline each metric should be applied to perform quality measurement. [20 marks]

When an appearance-based metric is employed, we are interested in the average sum of squared distances between all corresponding pixels of the images generated by the two polygon meshes. Although this seems to be a very simple method of comparison, it has been shown to be useful, as if one of the polygon meshes is a good approximation of the other thin this difference will be small. This comparison can be done over all possible views to fully measure the quality difference between the two polygon meshes. This is applied to the final image produced by the rendering pipeline, after the rasterisation and hidden-surface removal stages. This means that it is expensive to calculate.

When a geometric-based method is employed, it estimates the average and maximum deviations between two meshes by comparing points in the first mesh to the closest points in the second mesh using some distance measure. As this operates on the raw polygons, it means that it can be performed at the beginning of the rendering pipeline, and therefore the full rendering process is not needed, making this far less computationally expensive than the appearence-based metric. Performance of this metric can be further improved by only using a subset of points to calculate the similarity.

2 Explain how the Hausdorff distance can serve as a metric to determine the dissimilarity between two polygon meshes, even when these meshes are formed by different numbers of vertices and connectivity. [10 marks]

The Hausdorff distance is a commonly used geometric different measure, as it measures the maximum deviation between two models. It can achieve this even when the meshes have different numbers of vertices and connectivity because it does not try to directly match points, but instead compares points in the first mesh to the closest points in the second mesh. This means that the number of vertices and connectivity are not important, and one point in the first mesh can be the closest to multiple points in the second mesh and vice versa. The final distance that it returns, x, is the longest of any of these distances from a point in mesh 1 to the closest point in mesh 2, telling us that every point in mesh 1 is at most x distance from a point in mesh 2.

3 Describe the data structure of progressive meshes. Analyse the rendering efficiency of progressive meshes visualisation, given that the user is allowed to freely rotate the viewpoint during the visualisation process. [20 marks]

A progressive mesh represents polygon models as a sequence of edge collapses. There are two important advantages associated with this technique, that the resulting representation can be much simpler than the original model, and the time taken to reconstruct the model is directly proportional to the desired approximation size. These advantages are due to the fact the operations used are innately reversible, so we can store the simplest possible base mesh along with the sequence of operations used to get from the original mesh to this base mesh. A variable number of these edge collapses can then be reapplied in reverse as vertex splits to set the level of detail to be the same as desired.

However, for a large model such as a landscape, we will often only be looking at a small part of it, whereas a progressive mesh is view independent. This means that when we want high resolution, the whole model is in high resolution, and vice versa. Not being able to focus on a specific region means that we cannot remove as many polygons as we would like to for some of the views. This can be solved using the process of selective refinement to optimise the number of triangles, but this is difficult as there are string dependencies between neighbouring regions that would be affected by edge collapses and vertex splits.

4 Explain how progressive meshes implement the refinement and decimation processes. [10 marks]

A refinement algorithm begins with a initial coarse approximation and iteratively adds elements, where as a decimation algorithm begins with the original surface and iteratively removes elements down to the requested size. Both of these derive an approximation through a transformation. Progressive meshes implement refinement using vertex split operations, which divide a vertex into two new vertices, forming a new edge and two new triangles. Decimation is implemented using edge collapse operations, which take two connected edges and replace them with a single vertex, removing a new edge and two triangles.

5 Analyse how the incorporation of level-of-detail modelling impacts the rendering performance and network bandwidth consumption of a large distributed virtual environment system. Evaluate the suitability of using progressive meshes to implement the level-of-detail modelling in such a system.

Note that in the above distributed virtual environment system, all graphics models of the virtual environment are maintained by a remote server. During runtime, each client will download relevant graphics models on-demand from the server to support interaction and visualisation. [20 marks]

Using level-of-detail modelling decreases the network bandwidth consumption of a large distributed virtual environment system because it only the model of the appropriate resolution to serve each user request, taken from the geometry database stored on the server. This is clearly an improvement in comparison to transmitting the full model, or completely replacing it for each different quality level. Additionally, the majority of geometry compression methods take into account the information that is shared by neighbouring polygons, which reduces the amount of data that need to be transmitted to represent the full model. Level-of-detail models can also utilise progressive transmission, meaning that the models are encoded in such a way that partially transmitted models can be rendered and progressively refined as more information is received. This works based on the transmission of a base mesh and a list of transformations to be applied to it, so these transformations can progressively be applied to build up the full model. This means that the rendering process does not have to wait for all the information to arrive before it can begin to work on rendering the model, and therefore improves the rendering performance observed by the user.

6 Explain the main issue of applying level-of-detail modelling to support interactive visualisation of a large 3D scene, given that the user is allowed to change the viewpoint from time to time. Describe two different methods to tackle such a challenge. [20 marks]

An important issue to consider is the high polygon count that is required to display parts of meshes close to the observer at high visual quality, as well as lowering the level of detail for parts of meshes that are further away. This is very noticeable in situations such as rendering a large 3D landscape, when we are only interested in a small part of it.

View-dependent refinement can solve this issue, and this can be supported using a hierarchy of meshes. Parts of the scene outside of the view frustrum can be removed efficiently, and the level-of-detail model can be adjusted independently for each mesh in the hierarchy. Establishing such hierarchies on continuous surfaces is a challenging problem however, as it is difficult to correctly display the boundaries between meshes.

An alternative to this is criteria based refinement, which works based on the view frustum, surface orientation, and screen-space geometric error, and incrementally refines and coarsens the meshes in real time. This method effectively reduces the polygon count for large 3D scenes using minimal processing time, making it particularly effective for interactive visualisations.