1

A REVIEW ON LEVEL OF DETAIL

Tan Kim Heok, Abdullah Bade, Daut Daman

INTRODUCTION

Programmers have used Level of Detail (LOD) techniques to improve the performance and quality of their graphics systems since the mid nineteen-seventies. Numerous benefits can be obtained from the simplification of models, including reduced storage requirements, the reduction of computational complexity for scene rendering and fast transmission over network.

LEVEL OF DETAIL FRAMEWORK

Currently, there are four different kinds of LOD frameworks; discrete LOD, continuous LOD, view-dependent LOD and hierarchical LOD.

Discrete Level of Detail

Discrete level of detail is the traditional approach that creates LOD for each of the object separately during pre-process. At run-time, it picks each object's LOD according to the particular selection criterions. Therefore, it is called discrete LOD. The most significant advantage of discrete LOD is that it requires simple programming model. Secondly, it fits the modern graphics hardware well. Each level of detail easily can be compiled into triangle strips, display list, vertex array and so on. The rendering process is much faster than unorganized triangles on today's hardware. Even the implementation of discrete LOD is simple; however, it is not suitable for drastic simplification and not did scale well to large object.

Continuous Level of Detail

Continuous LOD which was developed in 1976 is a departure from the traditional discrete approach. As a contrary to discrete LOD, it creates data structure from which a desired level of detail can be extracted at run time. Objects do not use more polygons than necessary. Therefore, it has a better resource utilization and lead to better overall fidelity.

Smoother transitions can be created using continuous LOD since continuous LOD can adjust detail gradually and incrementally. It also can reduce visual pops. We can even geomorphic the fine-grained simplification operations over several frames to eliminate pops. Additionally, it supports progressive transmission.

View-Dependent Level of Detail

View-dependent LOD uses current view parameters to represent good quality of current view. A single object may thus spans several levels of detail. It is a selective refinement of continuous LOD. It shows nearby portions of object at higher resolution than distant portions. Silhouette regions of object are showed at higher resolution compared to interior regions do. View-dependent also take into account the user peripheral vision.

One of the advantages of view-dependent LOD is that it has a better granularity than continuous LOD. This is because it allocates polygons where they are most needed, within as well as among objects. It also enables drastic simplification of very large objects.

Hierarchical Level of Detail

View-dependent LOD solves the problem with large objects. However, it still faces difficulties in displaying small objects. Hence, hierarchy of LOD was created to solve the problem with small objects. It merges objects into assemblies. At sufficient distances, we also can create simplify assemblies, but not as an individual object.

Hierarchical LOD dovetails nicely with view-dependent LOD. It treats the entire scene as a single object to be simplified in view-dependent fashion. These discrete LOD will be grouped into a hierarchy and able to create better scalability for large structured models.

LEVEL OF DETAIL MANAGEMENT

Level of detail management is an important process in choosing the level of detail to represent each object. Traditionally, the system will assign a range of distances to each level of detail. Even though this method is extremely simple, it does not maintain constant frame rate. Moreover the correct switching distance may vary with the field of view and resolution. A more sophisticated level of detail management is enquired to enhance the LOD selection.

a. Size

An object's LOD is based upon its pixel size on the display device. It can overcome the weakness of distance selection criterion.

b. Eccentricity

An object's LOD is based on the degree to which it exists in the periphery of the display. Without a suitable eye tracking system, it is generally assumed that the user will be looking towards the centre of the display, so non-perceived objects are degraded.

c. Velocity

An object's LOD is based upon its velocity relative to the user. Funkhouser and Sequin (1993) acknowledge that the quick moving objects may appear blurred, or can be seen only for only a short period of time. Therefore the user may not be able to see them clearly. However, visual importance-biased image synthesis animation, which incorporates temporal changes into the models (Brown et al. 2003).

d. Fixed Frame Rate

An object's LOD is modulated in order to achieve and maintain a prescribed update rate. It is distinct from others because it is concerned with computational optimization rather than perceptual optimization. A combination of discrete and continuous approach is presented in time-critical rendering technique (Zach et al. 2002). It ensures acceptable frame rates even for complicated scenes. Therefore, it provides a convenient framework for real-time rendering applications.

e. Human Eyes Limitation

Resolution of element depends upon the depth of field that focused on the user's eyes where objects outside the fusional area appear in lower detail. Besides, visual disruptions, including eye saccade, flicker or blink are eyes' weakness. Saccade is a rapid reflex movement of the eye to fixate a target onto the fovea. Human do not appear to perceive detail during visual disruption occur. Change Blindness (Cater et al. 2003), where portions of the scene that have changed simultaneously with the visual disruption go unnoticed to the viewer. Attention is controlled entirely by slower, highly-level mechanisms in the visual system, which searches the scene, object by object, until attention finally focuses on the object that is changing.

There are two major influences on human visual attention: bottom-up and top-down processing. Bottom-up processing is the human automatic gaze direction for human to lively or colorful objects. In contrast, top-down processing is consciously directed attention to predetermined goals or tasks. This technique demonstrates the principle of Inattentional Blindness (Cater K. 2002), where portions of the scene unrelated to the specified task goes unnoticed.

f. Environment Conditions

Coarsen level of detail thresholds are determined through the use of haze, smoke, fog, clouds. These effects blur the scene and the actual detail hard to perceive.

g. Attention-Directed

Models of visual attention work out on where the user is likely to be looking. Visual attention-based technique allocates polygons to objects in a scene according their visual importance (Brown et al. 2003). The importance value is generated by considering objects' size, position, motion and luminance.

LEVEL OF DETAIL GENERATION

Refinement and decimation are the most common methodologies in surface simplification. Refinement algorithm begins with an initial coarse approximation and details are added at each step. Contrary to refinement, decimation algorithm begins the original surface and iteratively removes elements at each step. Both refinement and decimation share a very important characteristic: they seek to derive an approximation through a transformation of some initial surface.

Decimation simplification can be separate into two parts, either polygonal simplification or non-polygonal simplification. Non-polygonal simplification includes parametric spline surface simplification, simplification of volumetric models and also simplification of image based models. More works are done in polygonal simplification due to its flexibility and ubiquity. In fact, it is common to convert other model types into polygonal surfaces prior to processing. Practically, all virtual environment systems employ polygon renderers as their graphics engine.

Polygon simplification can be categorized into two parts; geometric simplification and topology simplification. Geometric simplification reduces the number of geometric primitives (vertices, edges, triangles). Topology simplification reduces the number of tunnels, holes and cavities. Aggressive simplification is a combination of geometric simplification and topology simplification. Based on Erikson C (1996), polygon simplification can be categorized as geometry removal (decimation), sampling and adaptive subdivision (refinement). Sampling is an algorithm that samples a model's geometry and then attempts to generate a simplified model that approximates the sampled data.

Geometric Simplification

a. Vertex Removal

Iteratively removing pieces of geometry perhaps is one of the most natural approaches in simplification. One such method is based on “pluck” vertices; each time a vertex and its incident triangles are removed, a hole is created, which must then be patched via triangulation (see Figure 1.1). Edge swap near the degenerated vertex prevents the folds in the mesh.

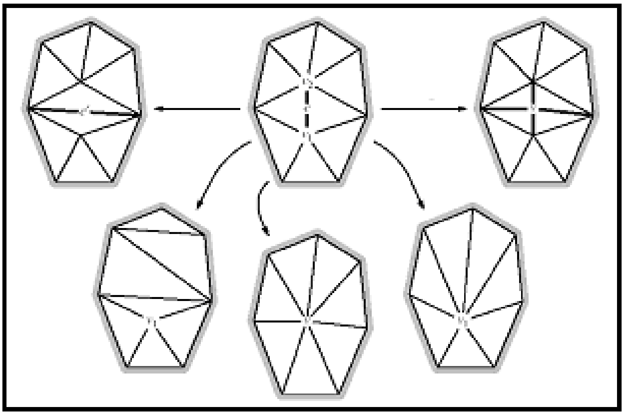

Figure 1.1 Local connectivity operations

The vertex removal method for arbitrary meshes was first introduced by (Schroeder et al. 1992). The decision whether to remove a vertex is based on the distance between the vertex and the plane. Vertices in flatter regions are preferred for removal.

"Simplification Envelopes" algorithm uses vertex removal as the coarsening operation (Cohen et al. 1996).

b. Vertex Clustering

The original *vertex clustering* approach for simplification is proposed by (Rossignac and Borrel 1993). The technique creates a uniform grid of rectilinear cells and only one vertex remained after simplification process. By optimizing the position of each cluster's vertex, the geometry can be improved. The "visual importance" of each vertex is rated by a simple heuristics and it was used to elect the representative vertex (Rossignac and Borrel 1993). Subsequently, a slight variation on this heuristic was motivated by introducing "floating cells" (Low and Tan 1997).

c. Edge Collapse

The *edge collapse* operation (Hope et al. 1993; Hope H. 1996) is a quite popular coarsening operation. The two edge's vertices are contracted to a single vertex, thereby deleting the edge and its incident triangles. The advantages are that the position of the substitute vertex can be chosen freely, can be optimized and no triangulation action is needed.

The general edge collapse algorithm involves two decisions; placing the substitute vertex and choosing the order of edges to collapse. In general, this can be done implicitly by specifying an error metric that depends on the position of the substitute vertex. Besides, other factors are involved, including topological constraints, geometric constraints, handling of degenerate cases and so on.

Half-edge collapse generally results in lower quality meshes than regular edge collapse since it allows no freedom in optimizing the mesh geometry, but has the advantage of having a more concise representation. *Triangle collapse* is yet another possible coarsening operation (Hamman 1994; Gieng et al. 1997). Vertices of a triangle are merged to a single new vertex. It provides little practical benefit.

A simple, fast and effective polygon reduction algorithm based on edge collapse (Wang and Ruan 2000). It utilizes the "minimal cost" method to calculate a set of LODs in 3D real time virtual environment. On the other hand, parallel triangular mesh decimation with the edge contraction can simplify object models in a short time without sorting (Franc and Skala 2001).

d. Vertex Pair Contraction

Vertex pair contraction is even more flexible than edge collapse in a way that it allows *any* pair of vertices to be merged, whether they share an edge or not. In order to limit the number of possible candidates for pair contraction, only a subset of pairs is considered. These pairs are called *virtual edges*, which are spatially close, are considered. Erikson and Manocha (1999) was the most outstanding for its dynamic selection of virtual edges, which allows increasingly larger gaps between pieces of a model to be merged.

e. Face Clustering

This dual of vertex clustering is less popular. This is due to the fact that the produced models generally exhibit relatively poor geometric and visual quality. The idea behind this approach is to merge nearly coplanar faces into large clusters of faces. Kalvin and Taylor (1996) refer to such clusters as "superfaces." The mesh is first partitioned into clusters. The interior vertices in each cluster are removed, and the cluster boundaries are simplified. Lastly, the resulting non-planar superclusters are triangulated. For others examples, see (Hinker and Hansen 1993; Garland 1999; Garland and Heckbert 1997).

Topology Simplification

For models with a large number of connected components and holes, it may be necessary to merge geometrically close pieces into individual larger ones to allow further coarsening (Erikson 2000). Algorithms based on vertex clustering and vertex pair contraction are by nature topology modifying (Lindstrom 2000). However, few of these algorithms are rather the byproduct of these coarsening operations. If preserving the manifoldness of a surface is important, then the topology simplification is applicable. However, manifoldness of the surface always less importance, and vertex pair contraction and its derivatives are adequate alternatives.

METRICS FOR SIMPLIFICATION AND QUALITY EVALUATION

The simplified model is rarely identical to the original, and therefore a metric is needed to measure how similar the two models are. If we are interested in the visual quality of a model, then an image metric may be more suitable. Such metrics have been developed, for example, to measure the degree of photorealism in computer generated images, search image databases, guide algorithms for image generation, and measure the image compression's quality.

Geometry-Based Metrics

Metrics for simplification are commonly used for two distinct purposes; evaluating the output quality, and determining where and how to simplify a model. If done correctly, a metric defined for simplification both determines the position of new vertices and the order in which coarsening operations are applied. However, it is sometimes difficult to express exactly what the metric is for a given simplification method.

The Hausdorff distance is probably the most well-known metrics for making geometric comparisons between two point sets. This metric is defined in terms of another metric such as the Euclidean distance. Quadric error metrics is based on weighted sums of squared distances (Garland and Heckbert 1997). The distances are measured with respect to a collection of triangle planes associated with each vertex. The beauty of the quadric metric is that it can be evaluated very efficiently by representing any set of planes as a single symmetric 4x4 matrix. This technique is fast and has good fidelity even for drastic reduction. Besides, it is robust in handling non-manifold surfaces. Aggregation in merging objects can also be performed here.

Vertex-vertex distance measures the maximum distance traveled by merging vertices. While vertex-plane distance store set of planes with each vertex, then errors are calculated based on distance from vertex to plane. Similarly, vertex-surface distance is the distance from vertex to surface. It maps point set to closest points to simplified surface. Maximum distance between input and simplified surfaces is used to measure surface-surface distance.

Attribute Error Metrics

Similar to geometry error metrics, it can be categorized into vertex-vertex distance, vertex-plane distance, vertex-surface distance and surface distance. Besides, it also include image-driven metric and perceptually-based metric. Some image metrics are rather simple and they treat each image in a geometric sense. Probably the most well-known metric for comparing images is the Lp pixel-wise norm, and in particular the $d2$ root mean square error. Perceptually-based metrics use contrast sensitivity function to guide simplification.

CURRENT ISSUES

In level of detail framework, the trends have shifted from traditional approach to view-dependent and hierarchical level of detail. Besides, human visual ability becomes an important issue in level of detail management (Reddy 1997).

A newly invented library by Cohen et al. (2003), named GLOD is a geometric level of detail system integrated into OpenGL rendering library. GLOD provides a low-level, lightweight Application Programming Interface (API) for level of detail operations. Hence, it is useful to simplified LOD developers' works.

There has been a big shift lately with out-of-core simplification models (Lindstorm P. 2000). Due to memory shortage in dealing with meshes that are significantly larger than available main memory, conventional simplification methods, which typically require reading and storing the entire model in main memory, cannot be used. As addition, graphics cards nowadays are able to render millions of triangles per second, hence, a truly scalable system for high-quality 3D interactive rendering of enormous data sets is wanted. Future work can also focus on network-streamed simplification. It has the potential to overcome the limit space in local hard disk problem. Another popular topic is artist-assisted LOD, where the user can control the degree of simplification based on user input.

CONCLUSION

The chapter presented most of the processes related to level of detail. It covers type of framework and LOD selection criteria that is suitable to the certain application domain. Future works are likely to focus more on out-of-core simplification and huge data memory management in various applications.

REFERENCE

BROWN, R., COOPER, L. AND PHAM, B. (2003B). Visual Attention-Based Polygon Level of Detail Management. Proceedings of the 1st international conference on Computer graphics and interactive techniques in Austalasia and South East Asia. *2003. New York, USA: ACM Press, 55-62.*

BROWN, R., PHAM, B. AND MAEDER, A. (2003A). Visual Importance-biased Image Synthesis Animation. Proceedings of the 1st international conference on Computer graphics and interactive techniques in Austalasia and South East Asia. *2003. New York, USA: ACM Press, 63-70.*

CATER, K. (2002). Selective Quality Rendering by Exploiting Inattentional Blindness: Looking but not Seeing. *Proceedings of the ACM symposium on Virtual reality software and technology*. 2002. New York, USA: ACM Press, 17-24.

Cater, K., Chalmers, A. And Dalton, C. (2003). Varying Rendering Fidelity by Exploiting Human Change Blindness. *Proceedings of the 1st international conference on Computer graphics and interactive techniques in Austalasia and South East Asia*. 2003. New York, USA: ACM Press, 39-46.

COHEN, J., LUEBKE, D., DUCA, N. AND SCHUBERT, B. (2003). *GLOD: Level of Detail for The Masses*. The Johns Hopkins University and the University of Virginia: Technical Report.

COHEN, J., VARSHNEY, A., MANOCHA, D., TURK, G., WEBER, H., AGARWAL, P. BROOKS, JR. F. P. AND WRIGHT, W. (1996). Simplification envelopes. In: Rishmeier, H. ed. *Proceedings of SIGGRAPH 96*. New Orleans, Louisianna: Addison Wesley. 119-128.

ERIKSON, C. (1996). *Polygonal Simplification: An Overview*. Department of Computer Science, University of North Carolina, Chapel Hill, NC: UNC Technical Report No. TR96-016.

ERIKSON, C. (2000). *Hierarchical Levels of Detail to Accelerate the Rendering of Large Static and Dynamic Polygonal Environments*. University of North Carolina at Chapel Hill: Ph.D. Thesis.

ERIKSON, C. AND MANOCHA, D. (1999). GAPS: General and Automatic Polygonal Simplification. In: Hodgins, J. and Foley, J. D. eds. *ACM symposium on Interactive 3D Graphics*. Atlanta, Georgia: ACM SIGGRAPH. 79-88.

FRANC, M. AND SKALA, V. (2001). Parallel Triangular Mesh Decimation without Sorting. 17th *Spring Conference on Computer Graphics (SCCG '01)*. April 2001. Budmerice,Slovakia: IEEE, 22-29.

FUNKHOUSER, T. A. AND SEQUIN, C. H. (1993). Adaptive Display Algorithm for Interactive Frame Rates During Visualization of Complex Virtual Environment. Proceedings of the 20th annual conference on Computer graphics and interactive techniques. 1993. New York, USA: ACM Press, 247-254.

GARLAND, M. (1999). *Quadric–Based Polygonal Surface Simplification*. Carnegie Mellon University: Ph.D. Thesis.

GARLAND, M. AND HECKBERT, P. S. (1997). Surface Simplification Using Quadric Error Metrics. In: Whitted, T. ed. *Proceedings* *of SIGGRAPH 97*. Los Angeles, California: ACM Press. 209-216.

GIENG, T. S., HAMANN, B. JOY, K. I., SCHLUSSMANN, G. L. AND TROTTS, I. J. (1997). Smooth Hierarchical Surface Traingulations, In: Yagel, R. and Hagen, H. eds. *IEEE Visualization '97*. Phoenix, Arizona: IEEE. 379-386.

HAMANN, B. (1994). A data Reduction Scheme for Triangulated Surfaces. *Computer Aided Geometric Design*. 11(20): 197-214.

HINKER, P. AND HANSEN, C. (1993). Geometric Optimization. In: Nielson, G. M. and Bergeron, D. eds. *IEEE Visualization '93*. San Jose, Carlifornia: IEEE Computer Society Press. 189-195.

HOPPE, H. (1996). Progressive Mesh. In: Rushmeier, H. ed. *Proceeding of SIGGRAPH 96. Computer Graphics Proceedings, Annual Conference Series*. New Orleans, Louisiana: Addison Wesley. 99-108.

HOPPE, H., DEROSE, T., DUCHAMP, T., MCDONALD, J. AND STUETZLE, W. (1993). Mesh Optimization. In: Kajiya, J. T. ed. *Proceeding of SIGGRAPH 93*. Anaheim, California: Addison Wesley. 19-26.

KALVIN, A.D. AND TAYLOR, R. H. (1996). Superfaces: Polygonal Mesh Simplification with Bounded Error. *IEEE Computer Graphics & Applications*. 16(3):64-77.

LINDSTROM, P. (2000). *Model Simplification using Image and Geometry-Based Metrics*. Georgia Institute of Technology: Ph.D Thesis.

LOW, K. L. AND TAN T. S. (1997). Model Simplification using vertex-clustering. In: Cohen, M. and Zeltzer, D. eds. *1997 ACM Symposium on Interactive 3D Graphics*. Phode Island: ACM SIGGRAPH. 75-82.

REDDY, M. (1997). *Perceptually Modulated Level of Detail for Virtual Environment.* University of Edinburgh: Ph.D Thesis.

RIBELLES, J., LÓPEZ, A., BELMONTE, O., REMOLAR, I. AND CHOVER, M. (2001), Variable Resolution Level-of-detail of Multiresolution Ordered Meshes *Proc. of 9-th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG 2001)*. February, 2001. Plzen: Czech Republic, 299-306.

ROSSIGNAC, J. AND BORREL, P. (1993). Multi-resolution 3d Approximations for Rendering Complex Scenes. In: Falciendo, B. and Kunii, T. L. eds. *Modeling in Computer Graphics*. Springer-Verlag. 455-465.

SCHROEDER, W. J., ZARGE, J. A. AND LORENSEN W. E. (1992). Decimation of Triangle Meshes. In: Catmull, E. E. ed. *Computer Graphics (Proceeding of SIGGRAPH 92)*. Chicago: Illinois. 65-70.

WANG, H. AND RUAN, Q. (2000). Fast Rendering of Complex Virtual Environment. *Proceedings of ICSP2000*. 1395-1398.

ZACH, C, MANTLER, S. AND KARNER, K. (2002). Time-critical Rendering of Discrete and Continuous Levels of Detail. *Proceedings of the ACM symposium on Virtual reality software and technology*. 2002. New York, USA: ACM Press.